

**Е. А. Глебова В. В. Крюкова**

**ИНФОРМАТИКА  
ЛАБОРАТОРНЫЙ ПРАКТИКУМ**

**Учебное пособие**

Кемерово 2020

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Кузбасский государственный технический университет  
имени Т. Ф. Горбачева»

Глебова Екатерина Андреевна  
Крюкова Валентина Валентиновна

**ИНФОРМАТИКА**  
**ЛАБОРАТОРНЫЙ ПРАКТИКУМ**  
**Учебное пособие**

Кемерово 2020

УДК 004(075.8)

Рецензенты

Директор института фундаментальных наук Кемеровского государственного университета, доктор технических наук, доцент Александр Михайлович Гудов

Директор института информационных и автоматизированных технологий Сибирского государственного индустриального университета, доктор технических наук, доцент Лариса Дмитриевна Павлова

Глебова, Е. А. **Информатика : лабораторный практикум** : учебное пособие / Е. А. Глебова, В. В. Крюкова ; Кузбасский государственный технический университет имени Т. Ф. Горбачева. – Кемерово, 2020. – 76 с. – ISBN 978-5-00137-170-0. – Текст : непосредственный.

Предназначено для выполнения лабораторных работ по дисциплине «Информатика». Учебное пособие содержит краткие сведения по MS Excel и MS Access, описание технологии выполнения лабораторных работ, контрольные вопросы и задания.

Разработано для технических направлений подготовки высшего образования (уровень бакалавриата).

Печатается по решению редакционно-издательского совета Кузбасского государственного технического университета имени Т. Ф. Горбачева

УДК 004(075.8)

© Кузбасский государственный технический университет имени Т. Ф. Горбачева, 2020

© Е. А. Глебова, В. В. Крюкова, 2020

ISBN 978-5-00137-170-0

## Оглавление

Введение .....	4
1. Основные инструменты Microsoft Excel .....	5
1.1. Лента .....	5
1.2. Математические функции MS Excel.....	8
1.3. Лабораторные работы .....	9
1.3.1. Табулирование функции одной переменной .....	9
1.3.2. Задание по лабораторной работе .....	19
1.3.3. Требования к содержанию и оформлению отчета .....	20
1.3.4. Контрольные вопросы.....	20
1.4. Табулирование параметрической функции .....	20
1.4.1. Задания по лабораторной работе .....	22
1.4.2. Требования к содержанию и оформлению отчета .....	25
1.4.3. Контрольные вопросы.....	26
1.5. Табулирование функции двух переменных .....	26
1.5.1. Задания по лабораторной работе .....	27
1.5.2. Требования к содержанию и оформлению отчета .....	30
1.5.3. Контрольные вопросы.....	31
1.6. Примеры выполнения лабораторных работ.....	31
2. Создание и использование базы данных MS Access .....	33
2.1. Логическое и физическое проектирование базы данных MS Access.....	33
2.1.1. Задание по лабораторной работе .....	54
2.1.2. Требования к содержанию и оформлению отчета.....	55
2.1.3. Контрольные вопросы.....	55
2.2. Примеры выполнения лабораторных работ .....	56
2.3. Создание основных объектов базы данных.....	58
Список рекомендуемой литературы .....	71

## Введение

Основными целями освоения дисциплины «Информатика» являются формирование навыков компьютерной обработки информации, владение основами библиографической культуры с применением информационно-коммуникационных технологий.

При изучении дисциплины обучающимся необходимо знать базовый курс информатики средней школы (основы работы в ОС Windows, основные понятия теории информации, основы работы с электронными таблицами и текстовом редактором, основы теории алгоритмизации и программирования).

Дисциплина «Информатика» является базовой для всех курсов, использующих автоматизированные методы анализа и обработки данных, а также компьютерную технику.

Дисциплина изучается на первом курсе в первом и втором семестрах. В первом семестре выполняются лабораторные работы в среде MS Excel, во втором – в среде MS Access.

Формой текущего контроля является предоставление и защита отчета по выполненным лабораторным работам.

Лабораторный практикум состоит из двух основных разделов: описание основных инструментов MS Excel и MS Access. Каждый из разделов включает описание технологии выполнения лабораторных работ, контрольные вопросы и задания.

# 1. Основные инструменты Microsoft Excel

## 1.1. Лента

Начиная с версии MS Excel 2013, вместо традиционного меню используется система логических групп и вкладок, называемая Лентой. Лента содержит ряд вкладок с командами, которые содержат средства для выполнения наиболее типичных задач в Excel.

Вкладка **Главная** (рис. 1) содержит самые основные команды, используемые при работе с MS Excel, такие как копирование, вставка, форматирование и стили. Она открывается по умолчанию при запуске Excel.

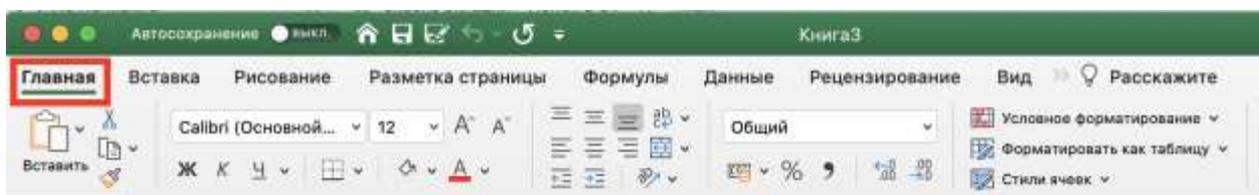


Рисунок 1 – Вкладка «Главная»

Вкладка **Вставка** (рис. 2) позволяет вставлять диаграммы, таблицы, фильтры и прочее, что дает возможность наглядно представить и визуализировать данные в Вашей рабочей книге.

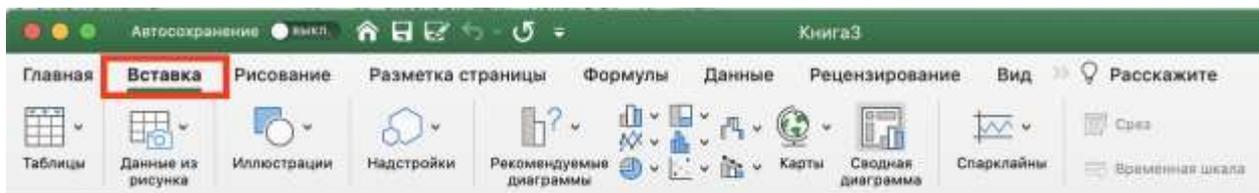


Рисунок 2 – Вкладка «Вставка»

Вкладка **Разметка страницы** (рис. 3) позволяет изменять форматирование книги, включая ширину полей, ориентацию листа и тему. Эти команды будут особенно полезны при подготовке книги к печати.

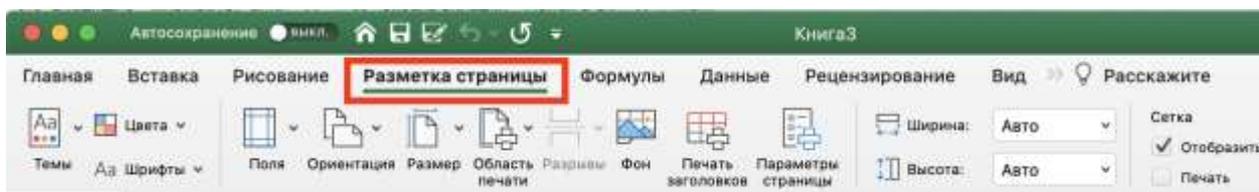


Рисунок 3 – Вкладка «Разметка страницы»

Вкладка **Формулы** (рис. 4) открывает доступ к наиболее часто используемым инструментам Excel – функциям и формулам. С помощью них Вы сможете вычислять и анализировать числовые данные, например, находить среднее значение или проценты.

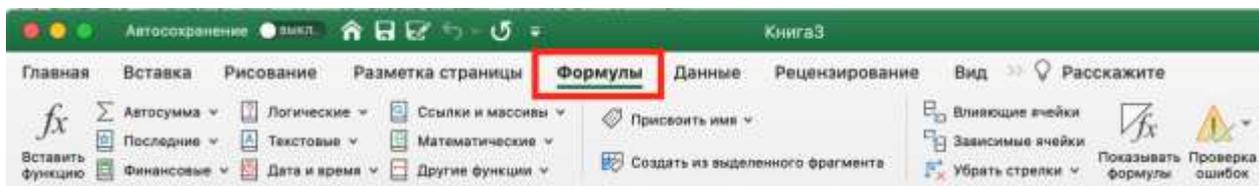


Рисунок 4 – Вкладка «Формулы»

Вкладка **Данные** (рис. 5) упрощает сортировку и фильтрацию информации в Вашей рабочей книге, что особенно удобно, когда книга Excel содержит большое количество данных.

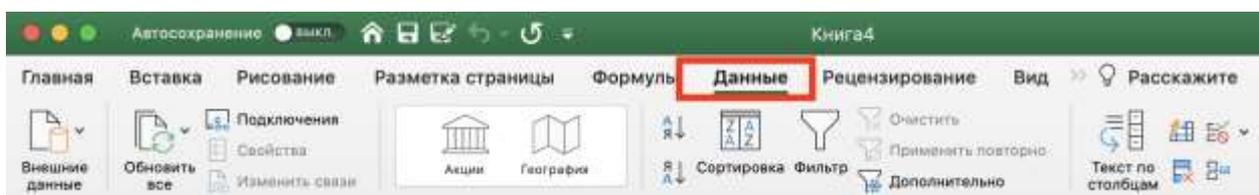


Рисунок 5 – Вкладка «Данные»

Вкладка **Рецензирование** (рис. 6) открывает широкие возможности редактирования текстов, включая добавление примечаний и отслеживание исправлений. Эти инструменты упрощают совместное использование и работу над книгами Excel.

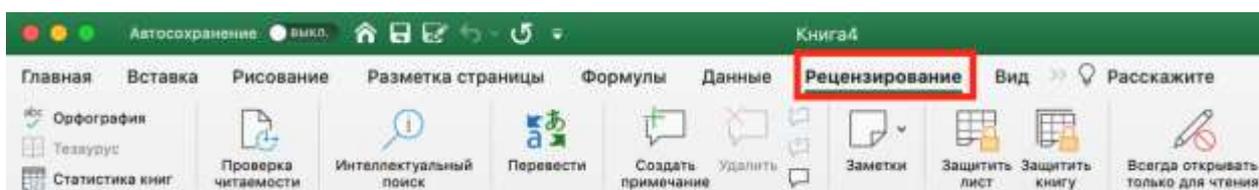


Рисунок 6 – Вкладка «Рецензирование»

Вкладка **Вид** (рис. 7) позволяет переключать режимы просмотра книги Excel и закреплять области для удобства работы с ними. Эти команды также необходимы при подготовке книги к печати.

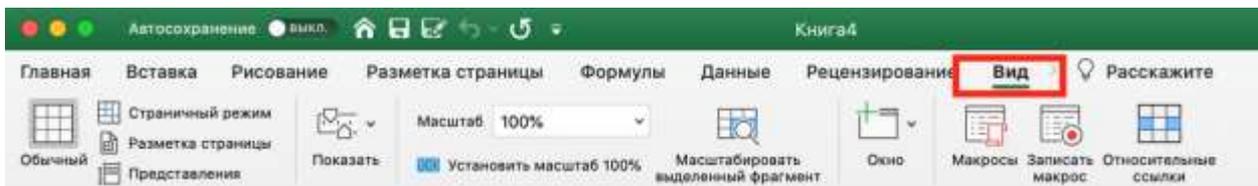


Рисунок 7 – Вкладка «Вид»

**Контекстные вкладки** (рис. 8) появляются на **Ленте** при работе с определенными элементами MS Excel, такими как таблицы или рисунки. Эти вкладки содержат особые группы команд, позволяющие форматировать данные элементы по мере необходимости.

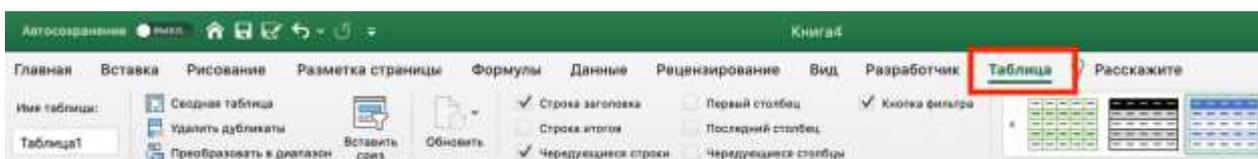


Рисунок 8 – Вкладка «Таблица»

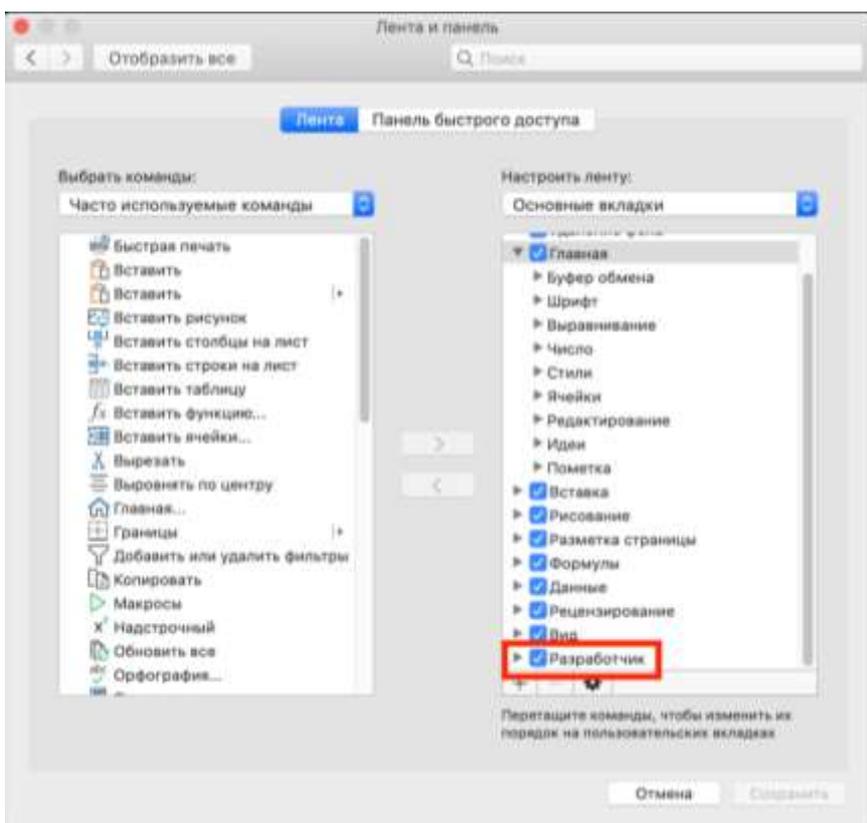


Рисунок 9 – Настройка ленты

Добавьте на **Ленту** вкладку **Разработчик**, если хотите создать макрос, экспортировать и импортировать XML-файлы или включить элементы управления. Чтобы добавить вкладку **Разработчик**, сделайте следующее:

1. На вкладке **файл** выберите **Параметры**, далее **Настройка ленты** (рис. 9).

2. В разделе **Настройка ленты** в списке *Основные вкладки* установите флажок **Разработчик**.

## 1.2. Математические функции MS Excel

Среди функций рабочего листа одну из наиболее многочисленных категорий образуют математические и тригонометрические функции. Применение этих функций позволяет значительно ускорить и упростить процесс вычислений. В качестве аргументов математических функций выступают, как правило, числовые значения (табл. 1).

Таблица 1 – Математические функции

Функция	Имя	Формула	Область определения
Модуль	ABS()	=ABS(2)	$-\infty < x < +\infty$
Арккосинус	ACOS()	=ACOS(-0,5)	$-1 \leq x \leq 1$
Арккотангенс	ACOT()	=ACOT(2)	$-\infty < x < +\infty$
Арксинус	ASIN()	=ASIN(-0,5)	$-1 \leq x \leq 1$
Арктангенс	ATAN()	=ATAN(1)	$-\infty < x < +\infty$
Косинус	COS()	=COS(60)	$-\infty < x < +\infty$
Котангенс	COT()	=COT(30)	$x \neq n\pi$
Логарифм	LOG()	=LOG(10)	$0 < x < +\infty$
Натуральный логарифм	LN()	=LN(2)	$0 < x < +\infty$
Степень	СТЕПЕНЬ()	=СТЕПЕНЬ(5;2)	$-\infty < x < +\infty$
Синус	SIN()	=SIN(ПИ()/2)	$-\infty < x < +\infty$

Создать макрос в MS Excel можно запустив средство записи макросов или с помощью редактора Visual Basic для приложений (VBA). После создания макроса следует сохранить его для возможности повторного использования в текущей или в других книгах.

Если понадобится использовать макрос только в той книге, где он был создан, нажмите кнопку **Сохранить** или **Сохранить как** (рис. 10) так же, как при обычном сохранении. Но сохранение книги с макросами происходит несколько иначе, поскольку необходимо использовать специальный формат файла «с поддержкой макросов».

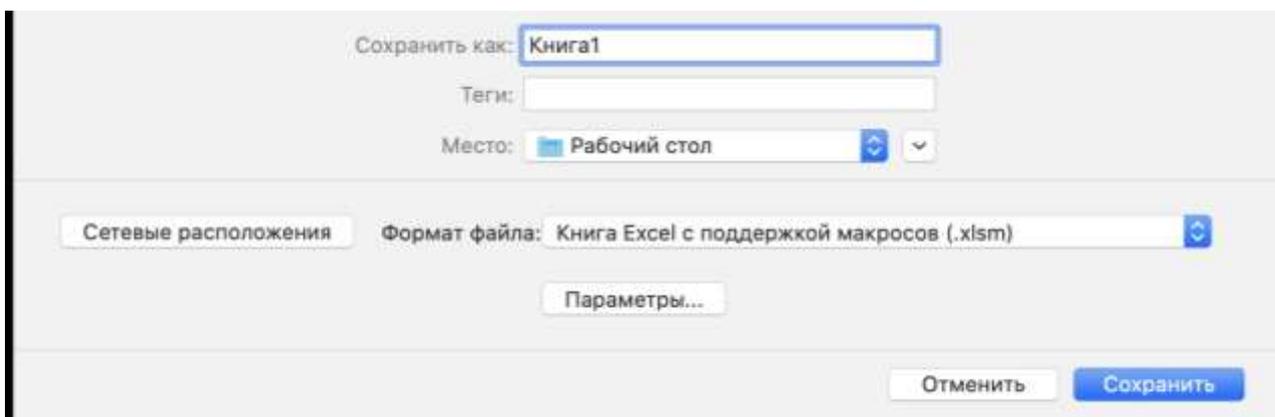


Рисунок 10 – Формат файла

### 1.3. Лабораторные работы

#### 1.3.1. Табулирование функции одной переменной

Табулирование функции начинается с ее математического анализа, в результате выполнения которого определяются область допустимых значений аргумента и наличие критических точек. Например, необходимо составить таблицу значений и построить график функции

$$Y = \frac{\sin(x) + \ln(1 + x)}{x}.$$

Рассматриваемая функция содержит натуральный логарифм  $\ln(x)$ , который определен только для положительных значений аргумента  $x > 0$ , то есть сама функция  $\ln(1 + x)$  определена только на отрезке  $x > -1$ . Кроме того, в точке  $x = 0$  имеется неопределенность типа  $0/0$  ( $\sin(0) = 0$ ,  $\ln(1) = 0$ ), раскрывая которую по правилу Лопиталя, получим

$$\lim_{x \rightarrow 0} \frac{\sin(x)}{x} = 1 \text{ — первый замечательный предел;}$$

$$\lim_{x \rightarrow 0} \frac{\ln(1 + x)}{x} = \frac{(\ln(1 + x))'}{x'} = \frac{1}{1 + x} = \frac{1}{1 + 0} = 1.$$

Отсюда при  $x = 0$  значение исследуемой функции равно 2.

Исходя из представленного анализа функции, вычисление ее значения для заданного значения аргумента можно представить следующим образом (рис. 11).

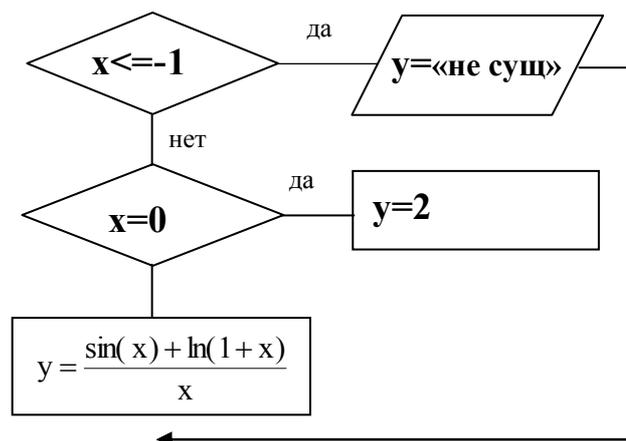


Рисунок 11 – Алгоритм вычисления значения функции

Рассмотрим последовательность действий для реализации табулирования указанной функции на рабочем листе MS Excel (рис. 12). Этот процесс, как правило, начинается с вывода заголовка работы, оформления шапки создаваемой таблицы и, для удобства изменения границ и величины шага табулирования функции, таблицы со значениями исходных данных: левой границы –  $a$ , правой границы –  $b$ , количества шагов табулирования –  $n$ , и величины шага табулирования –  $h$ , которая определяется по формуле  $h = (b - a) / n$ .

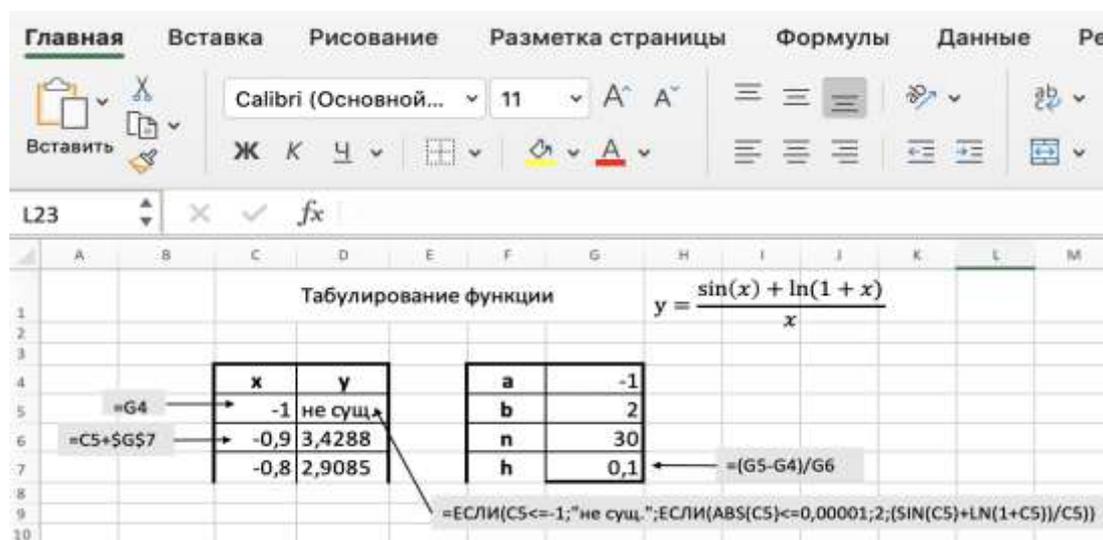


Рисунок 12 – Табулирование функции одной переменной

На приведенном рисунке 12 кроме образца оформления работы на рабочем листе MS Excel представлены и вычислительные формулы, в которых переменные, используемые в математических выражениях, заменены на адреса ячеек рабочего листа Excel, где хранятся значения соответствующих переменных. Так, например,

вышеприведенная формула определения величины шага табулирования, помещенная в ячейку **G7**, имеет вид:  $=(G5-G4)/G6$ , так как в **G5** хранится значение **b**, в **G4** – **a**, а в **G6** – **n**. Набор адресов ячеек при наборе формулы производится щелчком мышки по соответствующим ячейкам. Знак равно (=) в начале формулы, который в данном случае читается «присвоить», является признаком выполняемой формулы, поэтому после ее набора и нажатия клавиши **Enter** в ячейке **G7** будет выведен результат выполнения этой формулы. Если же при наборе формулы знак равно опустить, то набрать в формуле адреса ячеек щелчком мышки будет невозможно, а при ручном наборе этих адресов набранная формула будет воспринята как текст и выполнена не будет.

При наборе формул также необходимо учитывать способы адресации ячеек. В MS Excel их три: относительная, абсолютная и смешанная. Относительная адресация ячеек имеет вид:

**<имя столбца><номер строки>**.

Этот способ адресации ячеек уже был использован при записи формулы определения величины шага табулирования. Особенностью относительной адресации ячеек является изменение адреса указываемой ячейки при копировании формулы. Например, при копировании формулы в строке в копиях будет изменяться имя столбца, например:  $=B5 \rightarrow =C5 \rightarrow =D5$  и т.д.

При копировании же формулы в столбце, в копиях формулы будет изменяться номер строки. Если же возникает необходимость зафиксировать в формуле адрес какой-либо ячейки, то используется абсолютная адресация. Для этого после набора адреса ячейки, который необходимо зафиксировать, надо проследить, что курсор мышки находится в пределах фиксируемого адреса, и нажать на клавишу **F4**, после чего перед именем столбца и номером строки появится символ **\$**, который и является признаком абсолютной адресации. Например, набрана  $=B7$ , где курсор мышки, и нажата клавиша **F4**, при повторном нажатии клавиши будет получено  $=B\$7$ . При редактируемом адресе символ **\$** останется только перед номером строки:  $=B\$7$ . При третьем нажатии **F4** зафиксируется имя столбца, а номер строки становится относительным, то есть изменяющимся при копировании:  $=\$B7$ . При четвертом нажатии адрес ячейки снова станет относительным:  $=B7$ . Таким образом, при втором и третьем нажатии клавиши **F4** получается смешанная адресация ячейки рабочего листа MS Excel.

При работе с формулами и функциями в Excel используется несколько типов ссылок: относительные, абсолютные и смешанные. Чаще всего в формулах используют относительные ссылки.

**Относительная ссылка** – это ссылка на ячейку, меняющаяся при копировании формулы. Относительная ссылка на ячейку **A1** выглядит так: **A1**. Например, если в ячейку **A3** ввести формулу **=A1+A2**, то при копировании ее из ячейки **A3** в ячейку **B3** она преобразуется в формулу **=B1+B2**.

**Абсолютная ссылка** жестко привязывает формулу к конкретной ячейке и не изменяется при копировании. Чтобы сделать ссылку абсолютной, нужно добавить знак **\$** перед буквой и числом, которые составляют адрес ячейки, например **\$G\$7**. Это можно сделать вручную или нажав клавишу **F4** после ввода адреса ячейки.

**Смешанная ссылка** является комбинацией абсолютных и относительных ссылок. Например, если не должен меняться только столбец, то адрес записывается как **\$B5**; если не должна меняться только строка, то адрес записывается как **B\$5**.

Заполнение таблицы значений аргумента и функции, то есть сам процесс табулирования функции, начинается с занесения в эту таблицу значения левой границы табулирования **a**, для чего в ячейку, начиная с которой должны быть размещены значения аргумента, в данном примере это ячейка **B5**, заносится формула **=G4**. В следующую ячейку **B6** заносится формула **=B5+\$G\$7**, которая должна копироваться в колонке до тех пор, пока не будет получено значение правой границы табулирования функции. Поэтому первое слагаемое, указывающее предыдущее значение аргумента, должно изменяться, а вот второе слагаемое должно быть зафиксировано, так как оно указывает ячейку, в которой находится значение шага табулирования.

Копирование содержимого ячейки, в том числе и занесенной в нее формулы, производится следующим образом: левой клавишей мышки делается щелчок по выбранной ячейке, в результате чего она активизируется и ее границы выделяются жирной линией, а в нижнем правом углу этой ячейки появляется прямоугольник, который является маркером копирования (заполнения). Курсор мышки устанавливается на этот маркер и превращается в жирный крестик, после чего нажимается левая клавиша мышки и ее курсор протягивается в том направлении, в котором необходимо скопировать со-

держимое данной ячейки. В данном примере курсор мышки протягивается вниз.

Следующий этап табулирования функции – это запись формулы вычисления значения функции. В рассматриваемом примере алгоритм вычисления значения функции содержит две структуры ветвления, реализация которых требует использовать функцию ЕСЛИ, которая имеет вид

**ЕСЛИ(лог\_выражение;значение\_если\_истина;значение\_если\_ложь).**

Для удобства набора формул MS Excel имеет мощный аппарат функций рабочего листа, который содержит более 400 функций.

Для поиска и выбора нужной функции используется мастер функций, запуск которого производится щелчком мышки по кнопке *fx* в строке формул (рис. 13).

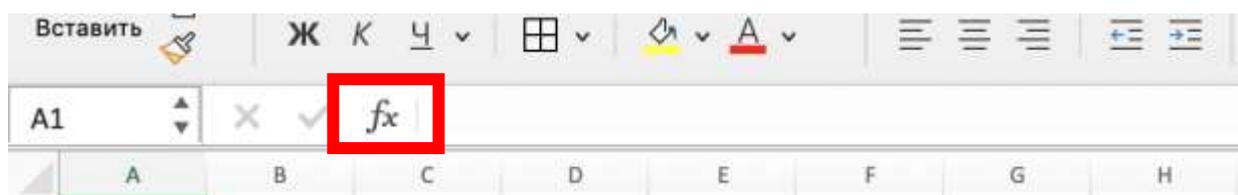


Рисунок 13 – Строка формул

В результате на фоне рабочего листа появится окно *Построителя функций* (рис. 14).

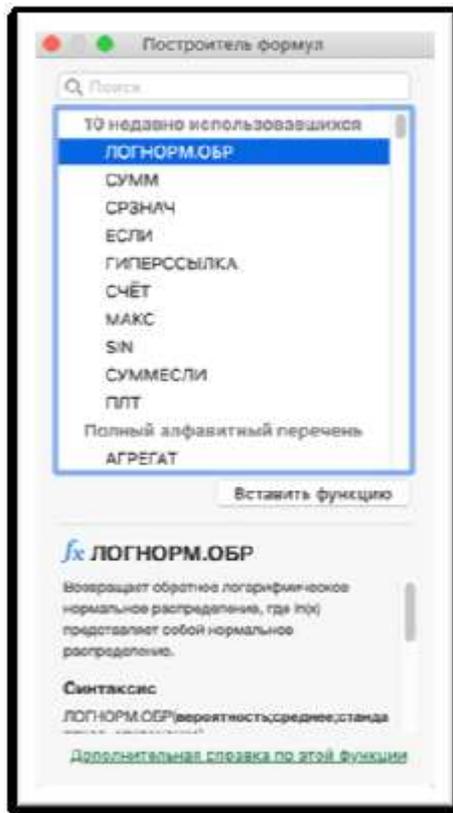


Рисунок 14 – Построитель функций

Поиск требуемой функции в мастере функций может быть выполнен следующими способами:

- в окне «10 недавно использовавшихся» просмотреть представленные в нем функции и, если имеется нужная вам функция, щелкнуть по ней мышкой, в результате чего она будет выделена;
- в окне «Поиск функции» набрать имя искомой функции, если оно вам хорошо известно, после чего щелкнуть мышкой по кнопке **Найти**. При правильном наборе имени функции оно появится в начале списка функций в окне «Выберите функцию» и автоматически будет выделено.

После того, как будет найдена и выделена нужная функция, необходимо щелкнуть мышкой по кнопке **Вставить функцию**, что приведет к появлению на экране шаблона этой функции. Указанная выше функция **ЕСЛИ** включена в категорию «Логические». Вид шаблона этой функции показан на рис. 15.

Так как формула набирается в ячейке **C5**, то есть напротив первого значения аргумента, которое находится в ячейке **B5**, то логическое выражение будет иметь вид: **B5<=-1**.

В строку «*Значение\_если\_истина*» согласно алгоритму заносится текст «не сущ.». Этот текст при наборе можно не заключать в кавычки, они появятся автоматически при переходе к набору следующего параметра функции, а если это был последний параметр, то при завершении набора функции. Кстати, если при наборе какой-либо функции один из параметров, представляющий, например, математическое выражение, окажется заключенным в кавычки, то это будет свидетельствовать о синтаксической ошибке в записи этого параметра, в связи с чем он был воспринят как текст.

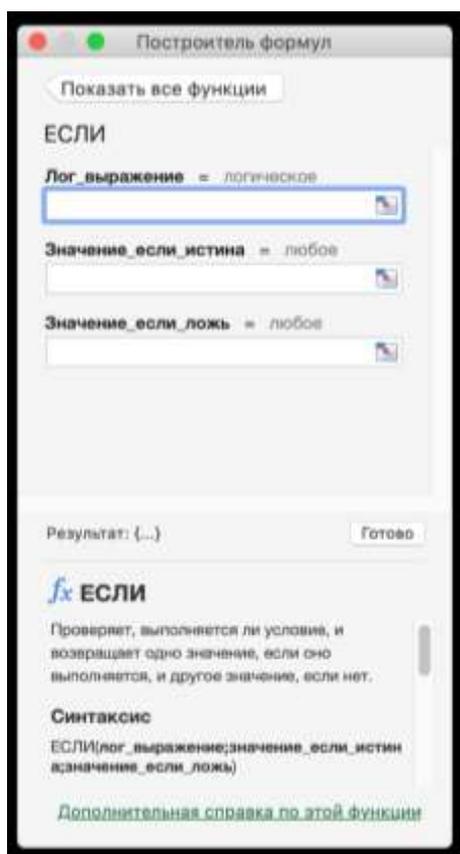


Рисунок 15 – Шаблон функции ЕСЛИ

Третий параметр «*Значение\_если\_ложь*» в рассматриваемом примере должен содержать функцию ЕСЛИ, то есть шаблон этой функции должен быть запущен вторично. Однако при незаконченном наборе одной функции произвести запуск шаблона другой или той же самой функции кнопкой *fx* невозможно. В этой ситуации, если необходимо вторично использовать набираемую функцию, можно щелкнуть мышкой по имени этой функции, которое выводится в самом начале строки формул (рис. 16).

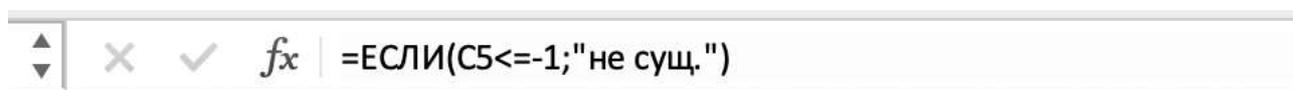


Рисунок 16 – Отображение набираемой функции в строке формул

В рассматриваемом примере формула для вычисления значения функции будет представлена следующим образом:

=ЕСЛИ(B5<=-1;"не  
сущ";ЕСЛИ(ABS(B5)<=0,00001;2;(SIN(B5)+LN(1+B5))/B5))

Необходимо отметить, что проверка значения аргумента на ноль в виде  $X=0$  практически не срабатывает, так как десятичная и

двоичная системы счисления кратными не являются и поэтому подавляющее большинство десятичных дробей в двоичную систему переводятся с погрешностью. Для компенсации этой погрешности проверку необходимо делать на ближайшие окрестности проверяемой величины. В данном случае эта проверка может быть записана следующим образом:  $|X| \leq 1e-5$ , что и было использовано при записи вышеприведенной функции.

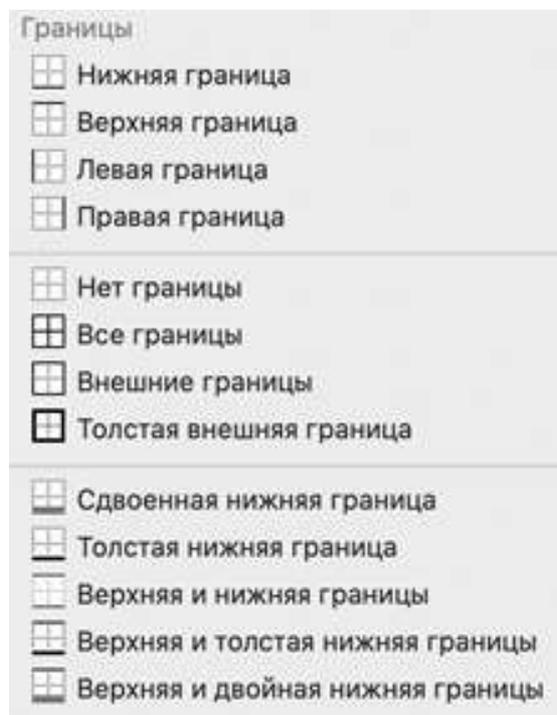


Рисунок 17 – Рисование границ

После набора табулируемой функции она копируется в соответствующем столбце таблицы до последнего значения аргумента, а затем с помощью мыши выделяются все ячейки таблицы и производится ее обрамление. Для этого необходимо щелкнуть мышкой по стрелке кнопки **Границы** – . В результате появится окно с кнопками рисования границ (рис. 17), используя которые и производят обрамление таблицы.

Вначале необходимо щелкнуть мышкой по кнопке  – **Все границы**, а затем по кнопке  – **Толстая внешняя граница**. Все ячейки таблицы при этом обязательно должны быть выделены. Обрамление таблицы завершается выделением ее шапки и щелчком мышкой по кнопке  – **Толстая внешняя граница**.

Построение графика функции начинается с выделения ячеек таблицы этой функции, содержащих значения аргумента и функции. Инструменты для работы с диаграммами собраны в группе **График** (рис. 18) на вкладке **Вставка**. Если щелкать мышью на кнопках с названиями типов графиков, будут раскрываться меню, где можно указать конкретную модификацию графика выбранного типа.



Рисунок 18 – Вставка графика

В данном случае график выглядит правильно, только в качестве названия диаграммы необходимости набрать реальное название диаграммы (рис. 19).



Рисунок 19 – Название диаграммы

Названия аргумента, легенды, функции и все необходимые настройки можно сделать во вкладке **Конструктор**, которая становится активной при нажатии на элемент **График** (рис. 20). В данном случае в названии осей достаточно набрать соответственно **X** и **Y**.

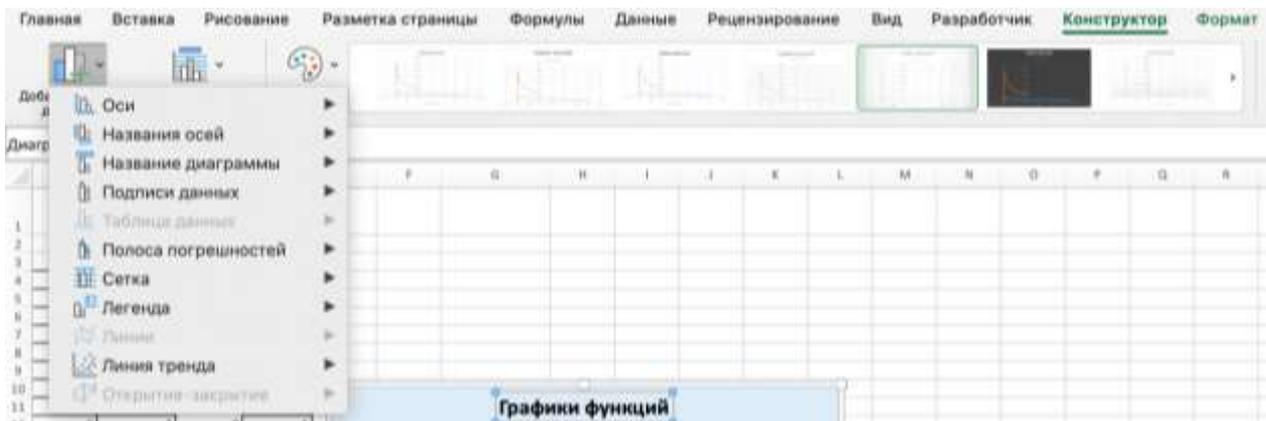


Рисунок 20 – Добавление и изменение данных на графике

В рассматриваемом примере диаграмма должна быть размещена на том же листе, где находится таблица (рис. 21).

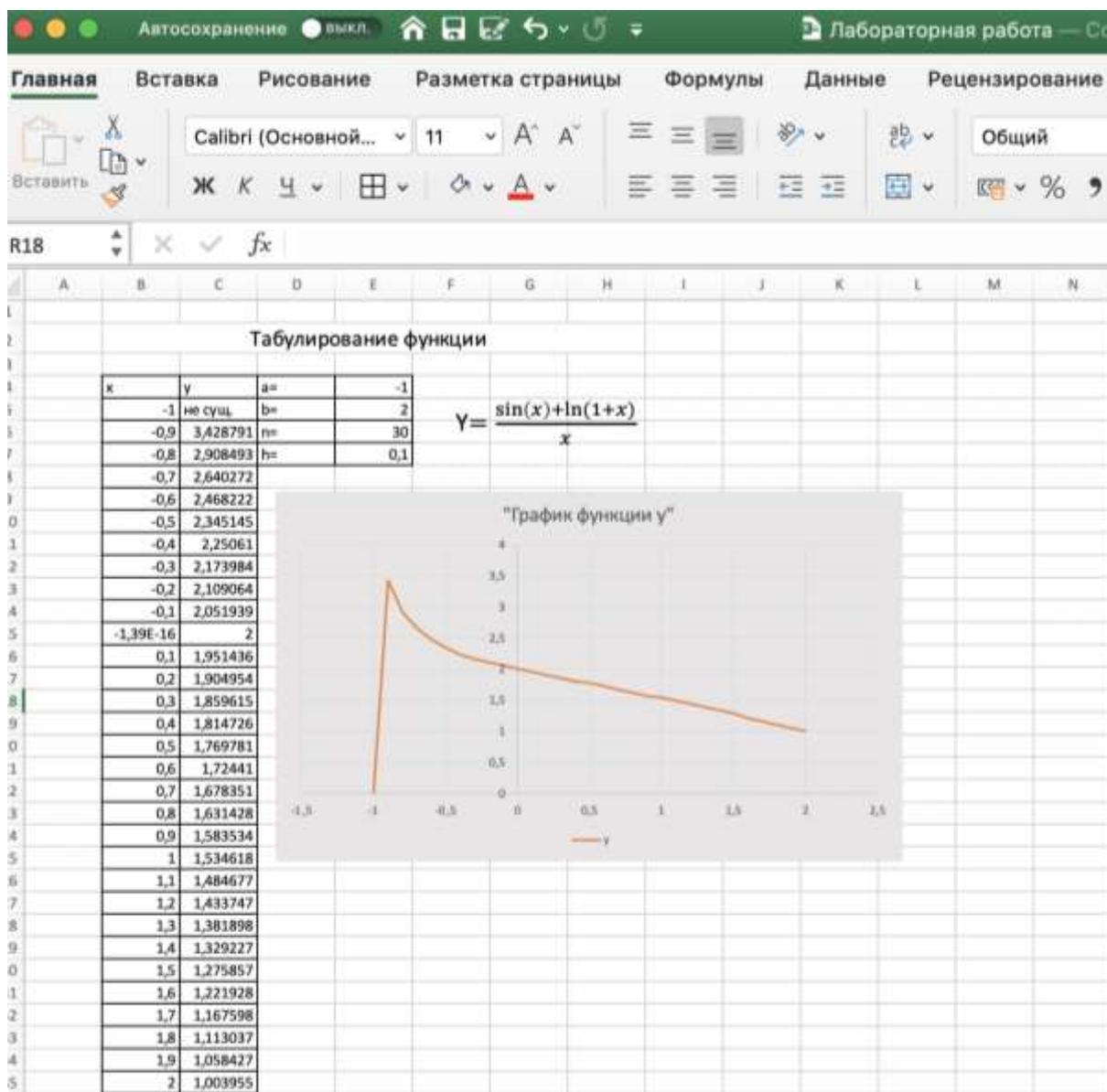


Рисунок 21 – Результат табулирования функции на заданном отрезке

### 1.3.2. Задание по лабораторной работе

Используя функцию из таблицы (табл. 2) и табличный процессор MS Excel, реализовать табулирование функции на основе рассмотренного примера и задания своего варианта, построить совмещенный график функций. При необходимости осуществить корректировку графика.

Варианты заданий приведены в таблице 2. Выберите варианты индивидуальных заданий в соответствии с вашим номером логина.

Таблица 2 – Задания для табулирования функций одной переменной

№	Функция $F(x)$	А	В	№	Функция $F(x)$	А	В
1.	$\frac{\sin(x)}{x} e^{-x}$	0	$\pi$	17.	$\sin(\ln(1+x)/x) e^x$	0	2
2.	$x^x e^{-x}$	0	2	18.	$x^5 e^{-x/\sin(x)}$	0	2
3.	$\frac{\operatorname{arctg}(x)}{x} e^{-x}$	0	1	19.	$\cos(2\arccos(x))$	-1	1
4.	$\frac{\ln(1+\pi x)}{x} e^{-x}$	0	2	20.	$\sin(2\arcsin(x))$	-1	1
5.	$x^{6,5} e^{-\cos(x)}$	0	2	21.	$\operatorname{tg}(2\operatorname{arctg}(x))$	0	0,5
6.	$\frac{(1-x)(1-x^2)}{1-x^5}$	0	2	22.	$\frac{\arcsin(x)}{x} e^{-x}$	0	1
7.	$\frac{\ln(1+x)}{x} - \frac{\sin^2(x)}{x^2}$	0	$\pi$	23.	$x^x e^{-x} \ln(1+x)$	0	1
8.	$\frac{\operatorname{tg}(x)}{x} e^{-x}$	0	$\pi/4$	24.	$\frac{xR}{R-1}$ $R = (1+x)^2$	0	1
9.	$\cos(3\arccos(x))$	-1	1	25.	$\frac{\sin(x)}{x} e^{-x} (1+x^2)$	0	2
10.	$\sin(3\arcsin(x))$	-1	1	26.	$x^3/(1-\cos^3 x)$	0	$\pi/2$
11.	$\operatorname{tg}(3\operatorname{arctg}(x))$	-0,5	0,5	27.	$10^{\ln(1+x)} \sin(x)$	0	1
12.	$10^{1-x} \frac{\ln(1+x)}{x}$	0	2	28.	$x^x \ln(x) e^{-x}$	0	3
13.	$\frac{xR}{R-1}$ $R = (1+x)^3$	0	1	29.	$\frac{1-x}{x} \lg(1+x)$	0	1
14.	$(x^{2,5} - x + 1) e^{x/2}$	0	2	30.	$\frac{\sin(x)}{x} e^{-x} (1+x^x)$	0	1
15.	$\frac{\sin(x)}{x} e^{-x} (1+x^3)$	0	1	31.	$\frac{\arcsin(x)}{x} e^{-x} \times$ $\times \ln(2+x)$	-1	1
16.	$x^x e^{-x^2}$	0	2	32.	$\frac{\arcsin(x) \ln(1+x)}{x^2}$	0	1

### 1.3.3. Требования к содержанию и оформлению отчета

Необходимо реализовать табулирование функции, рассмотренной в примере выше, и выполнить табулирование функции в соответствии с вариантом, результат представить на том же рабочем листе MS Excel (табл. 2).

Подготовить отчет по выполненной лабораторной работе средствами MS Word. Отчет должен содержать:

- краткое описание технологии выполнения работы;
- анализ полученных результатов;
- копию рабочего листа MS Excel в виде рисунков выполненной работы (задач с исходными данными и решениями);
- ответы на контрольные вопросы письменно в тетради «вопрос – ответ».

### 1.3.4. Контрольные вопросы

1. Ввод и редактирование данных в таблицах.
2. Встроенные функции. Назначение мастера функций.
3. Относительная, абсолютная и смешанная ссылка.
4. Функциональные возможности вкладок **Главная**, **Вставка** и **Разметка страницы**.
5. Вкладка **Формулы**.
6. **Контекстные** вкладки.
7. Определение понятия «табулирование функции».
8. Оформление таблицы.
9. Основное назначение электронных таблиц.
10. Укажите, что позволяет выполнять электронная таблица.
11. Назовите, как называется документ в программе MS Excel.
12. Укажите, в каком виде ячейка не может содержать данные.

### 1.4. Табулирование параметрической функции

Параметрической называется функция, имеющая такое представление переменных, когда их зависимость выражается через дополнительную величину – параметр.

В параметрическом задании функции, предназначенные для представления в декартовых координатах, представляемых в полярных координатах. Основным параметром этих функций является направление вектора, на котором откладывается отрезок. Параметрическая функция определяет координаты точки в декартовой системе

координат, в которой заканчивается этот отрезок. Например, параметрическое задание окружности имеет вид

$$x=R*\cos(t) \quad y=R*\sin(t),$$

где  $R$  – радиус окружности;  $t$  – направление вектора в радианной мере в диапазоне от 0 до  $2\pi$ .

Табулирование параметрических функций во многом похоже с табулированием функций с одной переменной, только исходным параметром, т.е. аргументом функции, является направление вектора, исходные значения которого и необходимо задать, а значения  $X$  и  $Y$  являются функциями этого аргумента. Исходя из сказанного, таблица табулирования параметрической функции имеет три колонки:  $t$ ,  $X$  и  $Y$ , а график функции строится по значениям  $X$  и  $Y$ .

На рисунке (рис. 22) значения исходных данных табулирования функции представлены в виде отдельной таблицы.

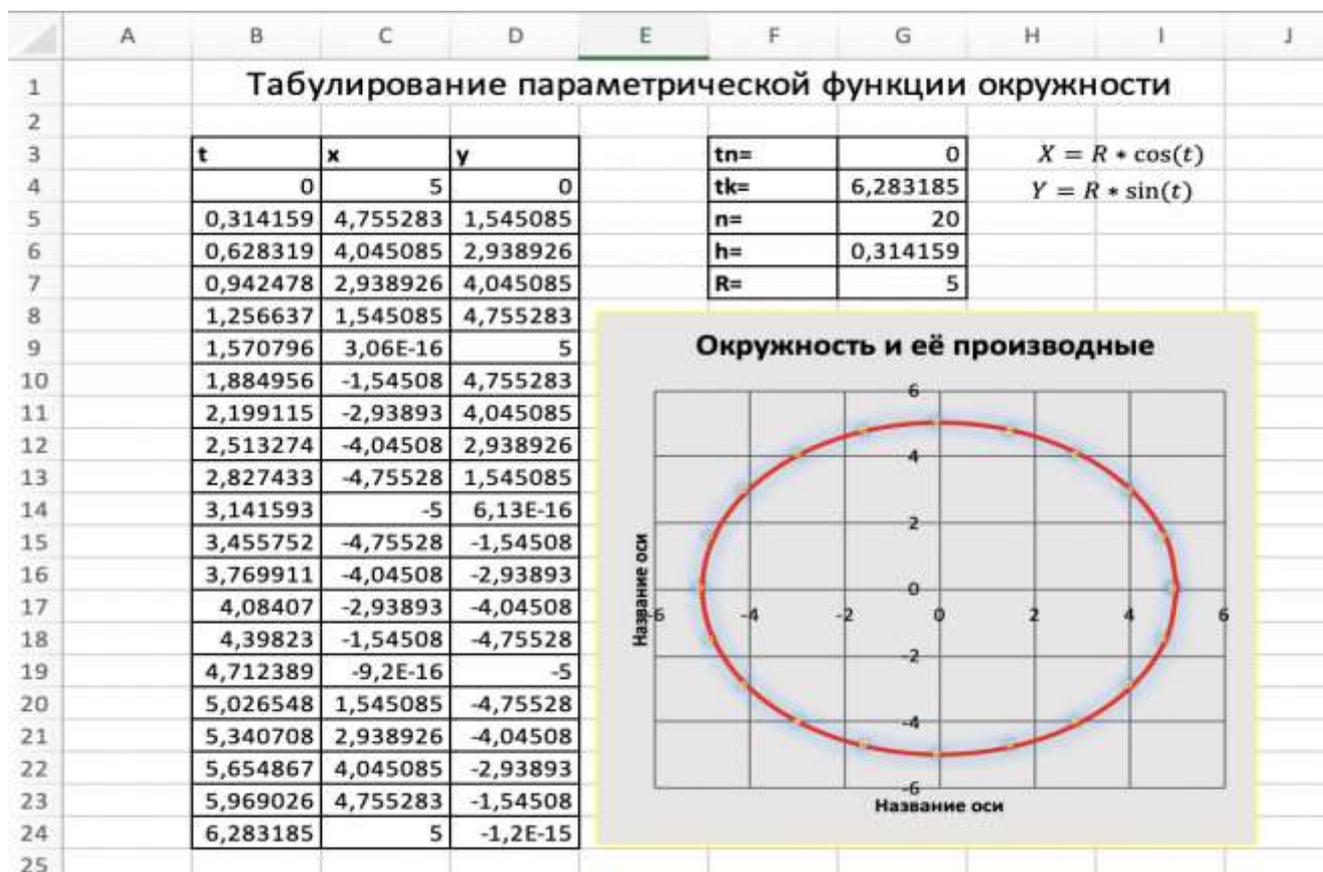


Рисунок 22 – Табулирование параметрической функции

В ней использованы следующие обозначения:  $tn$  – начальное направление вектора,  $tk$  – конечное направление вектора,  $n$  – количество изменений направления вектора,  $h$  – шаг изменения направления вектора,  $R$  – величина радиуса окружности. Конечное направление вектора  $tk$ , равное  $2\pi$ , здесь необходимо задать в виде

формулы:  $=2*\text{ПИ}()$ , где  $\text{ПИ}()$  – это функция, определяющая значение  $\pi$ , то есть 3,14159...

### 1.4.1. Задания по лабораторной работе

Используя функцию из таблицы (табл. 3) и табличный процессор MS Excel, реализовать табулирование параметрической функции на основе рассмотренного примера и задания своего варианта, построить графики функций. При необходимости осуществить корректировку графика. Варианты заданий приведены в таблице 3. Выберите варианты индивидуальных заданий в соответствии с вашим номером логина.

Таблица 3 – Задания на табулирование параметрической функции

Функция	Описание
1(16). Окружность и ее производные  Параметрическое задание: $x = R\cos(t)$ $y = R\sin(t)$ , где $t$ от 0 до $2\pi$ .	Начальное и конечное значения $t$ и шаг по $t$ задаются в радианной мере. При шаге по $t$ , существенно отличном от 0 и $2\pi$ и не равном $\pi$ , обеспечивается построение оригинальных звездоподобных фигур, многоугольников и орнаментов, вписанных в окружность. Для того, чтобы вычерчиваемые линии были замкнуты, подбирайте пределы и шаг по $t$ так, чтобы разность предельных значений была кратна $2\pi$ и величине шага.
2(17). Эллипс и его производные  Параметрическое задание: $x = A\cos(t)$ $y = B\sin(t)$ , где $t$ от 0 до $2\pi$ .	Начальное и конечное значения $t$ и шаг по $t$ задаются в радианной мере. При шаге по $t$ , существенно отличном от 0 и $2\pi$ и не равном $\pi$ , обеспечивается построение оригинальных звездоподобных фигур, многоугольников и орнаментов, вписанных в эллипс. Для того, чтобы вычерчиваемые линии были замкнуты, подбирайте пределы и шаг по $t$ так, чтобы разность предельных значений была кратна $2\pi$ и величине шага.
3(18). Астроида и ее производные  Параметрическое задание: $x = R\cos(t^3)$ $y = R\sin(t^3)$ , где $t$ от 0 до $2\pi$ .	Начальное и конечное значения $t$ и шаг по $t$ задаются в радианной мере. При шаге по $t$ , существенно отличном от 0 и $2\pi$ и не равном $\pi$ , обеспечивается построение оригинальных звездоподобных фигур и орнаментов. Чтобы вычерчиваемые линии были замкнуты, подбирайте пределы и шаг по $t$ так, чтобы разность предельных значений была кратна $2\pi$ и величине шага.
4(19). Розы и их производные  Параметрическое задание: $x = \sin(kt) + \sin((k-2)t)$	При нецелых $k$ возникает накладка лепестков со смещением. Шаг по $t$ должен быть достаточно малым. Предельное значение $t$ выбирается равным $\pi$ , умноженным на знамена-

Функция	Описание
$y = \cos(kt) - \cos((k-2)t)$	<p>тель числа <math>k/2</math>. Использование шага по <math>t</math>, существенно отличного от 0, порождает экстравагантные фигуры. Параметр <math>k</math> характеризует степень лепестковости: при четном целом <math>k-1</math> лепесток (<math>t</math> от 0 до <math>\pi</math>), при нечетном целом <math>2(k-1)</math> лепесток (1 от 0 до <math>2\pi</math>).</p>
<p>5(20). Кривые Хабенихта</p> <p>Параметрическое задание:  <math>x = r\cos(t)</math>  <math>y = r\sin(t)</math>  <math>r = A + B\cos(wt) + C\sin(wt)^2 + D\sin(wt)^4</math></p>	<p>Начальное и конечное значения <math>t</math> и шаг по <math>t</math> задаются в радианной мере. Параметр <math>w</math> характеризует степень лепестковости; параметры <math>A, B, C, D</math> – общую форму лепестков, их вытянутость и изрезанность, махровость цветка.</p> <p>Шаг по <math>t</math> должен быть достаточно малым. Предельное значение <math>t</math> выбирается равным <math>2\pi</math>, умноженным на знаменатель числа <math>w</math>. Использование шага по <math>t</math>, существенно отличного от 0, порождает экстравагантные фигуры.</p>
<p>6(21). Циклоида</p> <p>Параметрическое задание:  <math>x = rt - d\sin(t)</math>  <math>y = r - d\cos(t)</math></p>	<p>Начальное и конечное значения <math>t</math> и шаг по <math>t</math> задаются в радианной мере.</p> <p>Циклоида – траектория точки окружности радиуса <math>r</math>, катящейся по прямой линии. Если берется точка, отстоящая от центра окружности на расстоянии <math>d</math>, отличное от <math>r</math>, то возникает удлиненная (<math>d &gt; r</math>) или укороченная (<math>d &lt; r</math>) циклоида, называемая иногда трохойдой. Использование шага по <math>t</math>, существенно отличающегося от 0, порождает экстравагантные фигуры.</p>
<p>7(22). Эпитрохоида и ее производные</p> <p>Эпитрохоида в параметрическом задании:  <math>x = (R+r)\cos(t) - h\cos(t+mt)</math>  <math>y = (R+r)\sin(t) - h\sin(t+mt)</math>  <math>m = R/r</math>  <math>h</math> – расстояние от точки до центра катящейся окружности.          При <math>h = r</math> возникает эпициклоида.          При <math>h = R + r</math> возникает трохойдальная роза.          При <math>R = r</math> возникает улитка Паскаля.</p>	<p>Трохоиды представляют собой плоские кривые, являющиеся траекториями некоторой точки вне или внутри окружности радиуса <math>r</math>, катящейся по другой окружности радиуса <math>R</math>. Если катящаяся окружность имеет внешнее или внутреннее касание с неподвижной окружностью, то трохойда называется соответственно эпитрохоидой или гипотрохоидой. Предельные значения <math>t</math> подбираются так, чтобы их разность, умноженная на <math>m+1</math>, была кратна четному количеству <math>\pi</math>. Шаг по <math>t</math> должен быть достаточно малым (иначе получаем нечто экзотическое).</p>
<p>8(23). Гипотрохоида и ее производные</p> <p>Гипотрохоида в параметрическом задании:  <math>x = (R-r)\cos(t) + h\cos(mt-t)</math>  <math>y = (R-r)\sin(t) - h\sin(mt-t)</math>  <math>m = R/r</math>  <math>h</math> – расстояние от точки до центра катящейся окружности.          При <math>h = r</math> возникает гипоциклоида.</p>	<p>Трохоиды представляют собой плоские кривые, являющиеся траекториями некоторой точки вне или внутри окружности радиуса <math>r</math>, катящейся по другой окружности радиуса <math>R</math>. Если катящаяся окружность имеет внешнее или внутреннее касание с неподвижной окружностью, то трохойда называется соответственно эпитрохоидой или гипотрохоидой.</p>

Функция	Описание
<p>При <math>R = 2r</math> возникает эллипс.</p>	<p>Предельные значения <math>t</math> подбираются так, чтобы их разность, умноженная на <math>m - 1</math>, была кратна четному количеству <math>\pi</math>. Шаг по <math>t</math> должен быть достаточно малым (иначе получаем нечто экзотическое).</p>
<p>9(24). Эпициклоида и ее производные</p> <p>Эпициклоида в параметрическом задании:  <math>x = (R + r)\cos(t) - r\cos(t + mt)</math>  <math>y = (R + r)\sin(t) - r\sin(t + mt)</math>  <math>m = R/r</math>.</p>	<p>Эпициклоида представляет собой плоскую кривую, представляющую траекторию точки окружности радиуса <math>r</math>, катящейся по другой окружности радиуса <math>R</math> и имеющей с ней внешнее касание.</p> <p>В зависимости от <math>m</math> получаются кривые различной формы. При <math>m = 1</math> – кардиоида, при целых <math>m</math> кривая имеет <math>m</math> непересекающихся ветвей. Если точка лежит вне катящейся окружности, то кривая называется эпитрохоидой.</p> <p>Предельные значения <math>t</math> подбираются так, чтобы их разность, умноженная на <math>m + 1</math>, была кратна четному количеству <math>\pi</math>. Шаг по <math>t</math> должен быть достаточно малым (иначе получаем нечто экзотическое).</p>
<p>10(25). Гипоциклоида и ее производные</p> <p>Гипоциклоида в параметрическом задании:  <math>x = (R - r)\cos(t) + r\cos(mt - t)</math>  <math>y = (R - r)\sin(t) - r\sin(mt - t)</math>  <math>m = R/r</math>  при <math>m = 4</math> – астроида  при <math>m = 3</math> – кривая Штейнера.</p>	<p>Гипоциклоида представляет собой плоскую кривую, представляющую траекторию точки окружности радиуса <math>r</math>, катящейся по другой окружности радиуса <math>R</math> и имеющей с ней внутреннее касание.</p> <p>В зависимости от <math>m</math> получаются кривые различной формы: в случае целых <math>m</math> кривая имеет <math>m</math> непересекающихся ветвей. Если точка лежит вне катящейся окружности, то кривая называется гипотрохоидой.</p> <p>Предельные значения <math>t</math> подбираются так, чтобы их разность, умноженная на <math>m - 1</math>, была кратна четному количеству <math>\pi</math>. Шаг по <math>t</math> должен быть достаточно малым (иначе получаем нечто экзотическое).</p>
<p>11(26). Улитка Паскаля</p> <p>Параметрическое задание:  <math>x = \cos(t)(A\cos(t) + L)</math>,  <math>y = \sin(t)(A\cos(t) + L)</math>,  <math>t</math> от 0 до <math>2\pi</math>  <math>L &gt; 0</math>  <math>A = L</math> – кардиоида.</p>	<p>Начальное и конечное значение <math>t</math> и шаг по <math>t</math> задаются в радианной мере. Плоская кривая 4-го порядка, конхоида окружности радиуса <math>A</math>. Частный случай Декартова овала и эпитрохоиды.</p>
<p>12(27). Конхоида Никомеда</p> <p>Параметрическое задание:  <math>x = \cos(t)(A/\cos(t) \pm L)</math>  <math>y = \sin(t)(A/\cos(t) \pm L)</math>  <math>t</math> от <math>-\pi/2</math> до <math>\pi/2</math>.</p>	<p>Начальное и конечное значение <math>t</math> и шаг по <math>t</math> задаются без запроса к пользователю.</p> <p>Плоская кривая 4-го порядка; конхоида прямой <math>x = A</math>. Имеет две ветви с асимптотой <math>x = A</math>.</p>

Функция	Описание
<p>13 (28). Декартов лист</p> <p>Параметрическое задание:  <math>x = R\cos(t)</math> <math>y = R\sin(t)</math>  где <math>R = 3/2A\sin(2t) / (\cos(t)^3 - \sin(t)^3)</math>  <math>t</math> изменяется в границах от <math>-\pi/4</math> до <math>3\pi/4</math>  <math>x^3 + y^3 = 3Axy</math>.</p>	<p>Плоская кривая 3-го порядка, симметричная относительно биссектрисы <math>y = x</math> с асимптотой <math>y = -x - A</math>.</p>
<p>14 (29). Лемниската Бернулли</p> <p>Параметрическое задание:  <math>x = \pm R\cos(t)</math>  <math>y = R\sin(t)</math>  где <math>R = \sqrt{2 \cos(2t) }</math>  <math>t</math> изменяется от <math>-\pi/4</math> до <math>\pi/4</math>  <math>2A(x^2 + y^2) = 2A^2(x^2 - y^2)</math>.</p>	<p>Плоская кривая 4-го порядка. Начальное и конечное значение <math>t</math> и шаг по <math>t</math> задаются без запроса к пользователю.</p>
<p>15 (30). Овалы Кассини</p> <p>Уравнение в декартовых координатах имеет вид:  <math>(x^2 + y^2)^2 - 2C^2(x^2 - y^2) = A^4 - C^4</math>.  При <math>A &gt; C\sqrt{2}</math> – овальная линия.  При <math>C &lt; A &lt; C\sqrt{2}</math> – линия с талией.  При <math>A &lt; C</math> – два овала.  При <math>A = C</math> – лемниската Бернулли. В параметрическом задании  <math>x = R\cos(t)</math> и <math>y = R\sin(t)</math>,  где <math>R</math> – корни уравнения  <math>R^4 - 2C^2R^2\cos(2t) = A^4 - C^4</math>  <math>t</math> меняется в пределах от <math>-\pi/2</math> до <math>\pi/2</math>.</p>	<p>Плоская алгебраическая кривая 4-го порядка. Овал Кассини – это множество точек, произведение расстояний которых до 2 заданных полюсов <math>(-C,0)</math> и <math>(C,0)</math> является постоянной величиной.</p>

### 1.4.2. Требования к содержанию и оформлению отчета

Необходимо реализовать табулирование параметрической функции, рассмотренной в примере выше и выполнить табулирование параметрической функции в соответствии с вариантом, результат представить на том же рабочем листе MS Excel.

Подготовить отчет по выполненной лабораторной работе средствами MS Word. Отчет должен содержать:

- краткое описание технологии выполнения работы;
- анализ полученных результатов;
- копию рабочего листа MS Excel в виде рисунков выполненной работы (задач с исходными данными и решениями);
- ответы на контрольные вопросы письменно в тетради «вопрос – ответ».

### 1.4.3. Контрольные вопросы

1. Работа с графиком. Легенда. Подпись осей.
2. Редактор формул.
3. Параметрическая функция, определение.
4. Макросы. Запись макроса.
5. Назовите способы обращения к ячейке, расположенной на другом листе текущей книги.
6. Укажите, чем относительный адрес отличается от абсолютного адреса.
7. Назовите символ, который необходимо использовать, чтобы закрепить индекс адреса ячейки.
8. Как называется область вкладки, на которой располагаются функциональные иконки?
9. Укажите возможные типы данных ячейки Excel.
10. Укажите, с какого символа должна начинаться любая формула в MS Excel.

### 1.5. Табулирование функции двух переменных

Графически функция двух переменных представляет собой поверхность, поэтому ее табличное представление является матрицей, в первой строке которой размещаются значения переменной  $X$ , а в первом столбце – значения переменной  $Y$ , в остальные ячейки матрицы заносятся значения функции  $Z=f(X,Y)$ . Так как значения  $X$  находятся в одной строке, а значения  $Y$  в одном столбце, то в формуле вычисления значения функции  $Z$  в адресе  $X$  необходимо зафиксировать номер строки, а в адресе  $Y$  – имя столбца (см. в строке формул).

Например, необходимо построить график функции  $Z=X\cos(Y)$  для значений  $X$  и  $Y$  от  $-\pi$  до  $\pi$ . Шаг изменения  $X$  и  $Y$  примем равным  $\pi/5$  (рис. 23).

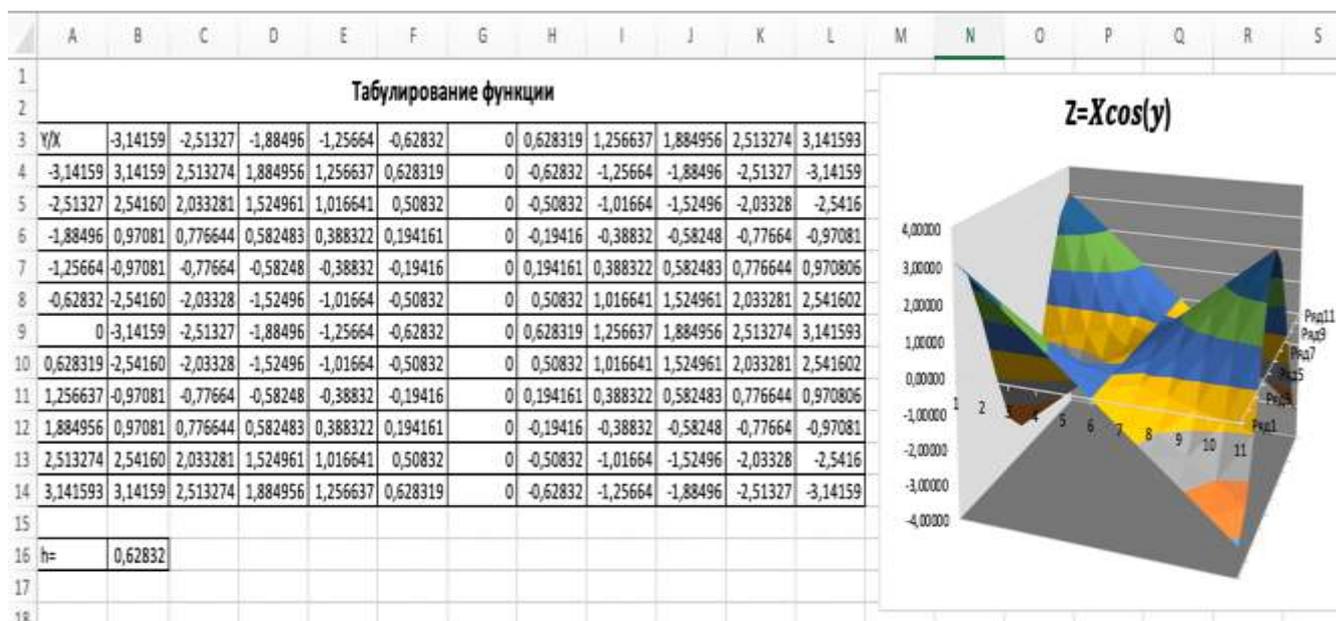


Рисунок 23 – Табулирование функции двух переменных

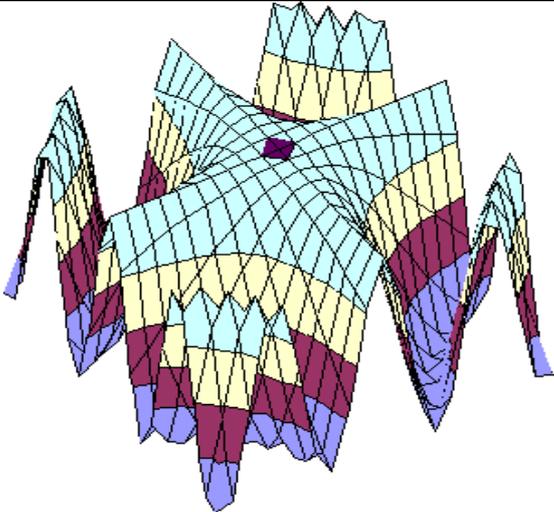
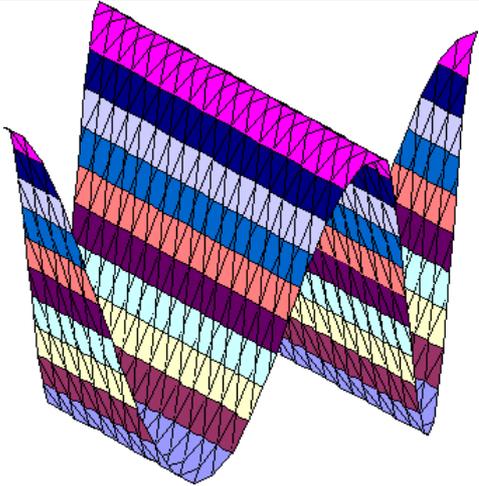
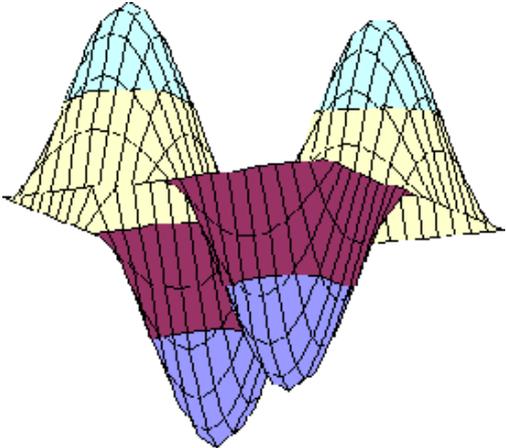
### 1.5.1. Задания по лабораторной работе

Используя функцию из таблицы (табл. 4) и табличный процессор MS Excel, реализовать табулирование функции двух переменных на основе рассмотренного примера и задания своего варианта, построить графики функций. При необходимости осуществить корректировку графика.

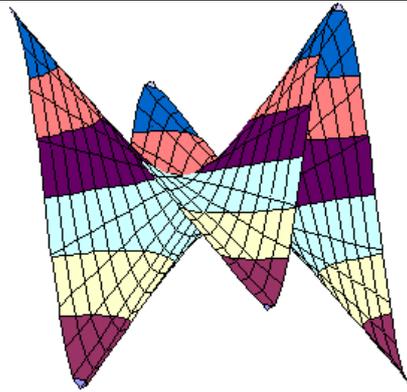
Варианты заданий приведены в таблице 4. Выберите варианты индивидуальных заданий в соответствии с вашим номером логина.

MS Excel позволяет изменить размеры рабочей области диаграммы посредством маркеров изменения размеров и ориентировать поверхность в пространстве.

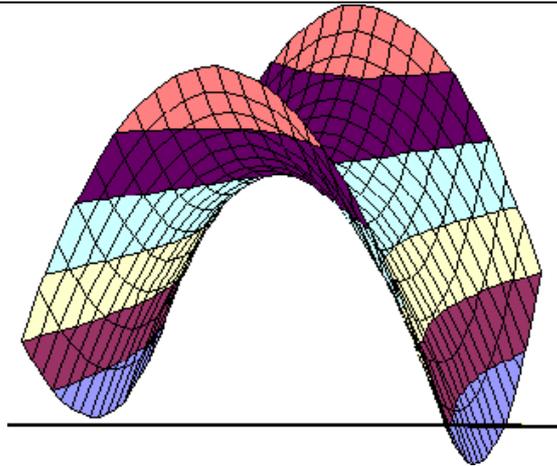
Таблица 4 – Задания – табулирование функций двух переменных

Номер варианта и аналитическое выражение функции	График
<p>1, 10, 19  <math>z = \cos(xy)</math></p>	
<p>2, 11, 20  <math>z = \cos(x + y)</math></p>	
<p>3, 12, 21  <math>z = \sin(x) \sin(y)</math></p>	

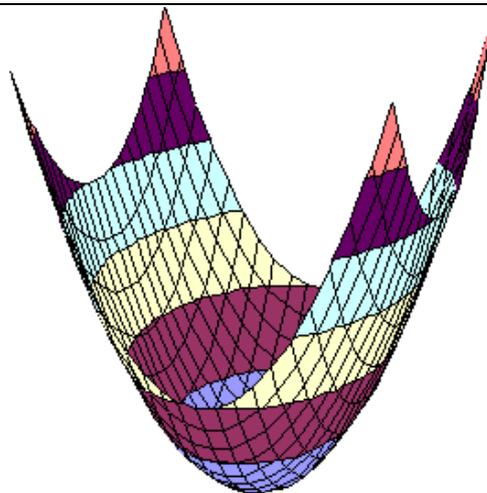
4, 13, 22  
 $z = x \cos(y)$

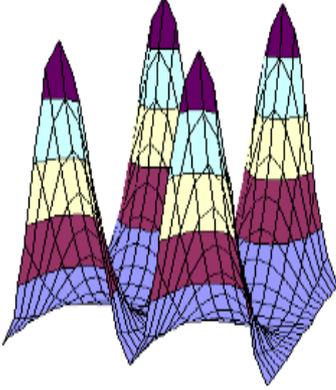
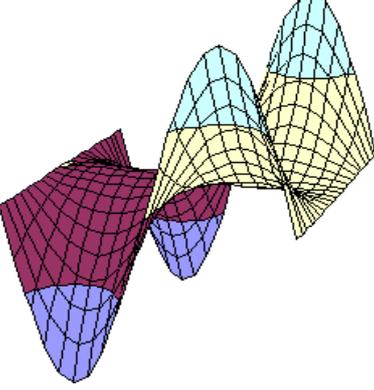
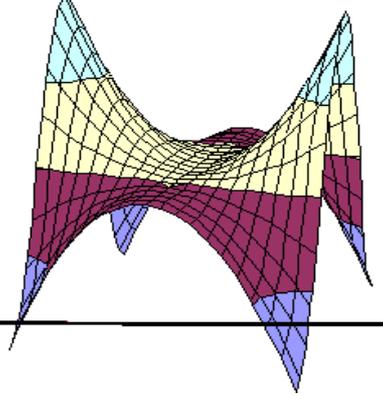


5, 14, 23  
 $z = x^2 - 2y^2$



6, 15, 24  
 $z = x^2 + 2y^2$



<p>7, 16, 25  <math>z = x^2 \sin(y)^2</math></p>	
<p>8, 17, 26  <math>z = x^2 \sin(y)</math></p>	
<p>9, 18, 27  <math>z = \cos(x)y^2</math></p>	

### 1.5.2. Требования к содержанию и оформлению отчета

Необходимо реализовать табулирование функции двух переменных, рассмотренной в примере выше, и выполнить табулирование функции двух переменных в соответствии с вариантом, результат представить на том же рабочем листе MS Excel.

Подготовить отчет по выполненной лабораторной работе средствами MS Word. Отчет должен содержать:

- краткое описание технологии выполнения работы;
- анализ полученных результатов;

- копию рабочего листа MS Excel в виде рисунков выполненной работы (задач с исходными данными и решениями);
- ответы на контрольные вопросы письменно в тетради «вопрос – ответ».

### 1.5.3. Контрольные вопросы

1. Укажите, какая ячейка называется активной.
2. Приведите пример, что не является типовой диаграммой в таблице.
3. Укажите, к какой категории относится функция ЕСЛИ.
4. Перечислите основные типы данных в MS Excel.
5. Укажите, что означает сообщение #знач! при вычислении формулы.
6. Назовите типы диаграмм, которые позволяют строить табличный процессор.
7. Перечислите типы фильтров в MS Excel.
8. Назовите аргументы функции ЕСЛИ().
9. Укажите последовательность действий при построении диаграммы типа «Поверхность».
10. Укажите алгоритм решения задачи.

## 1.6. Примеры выполнения лабораторных работ

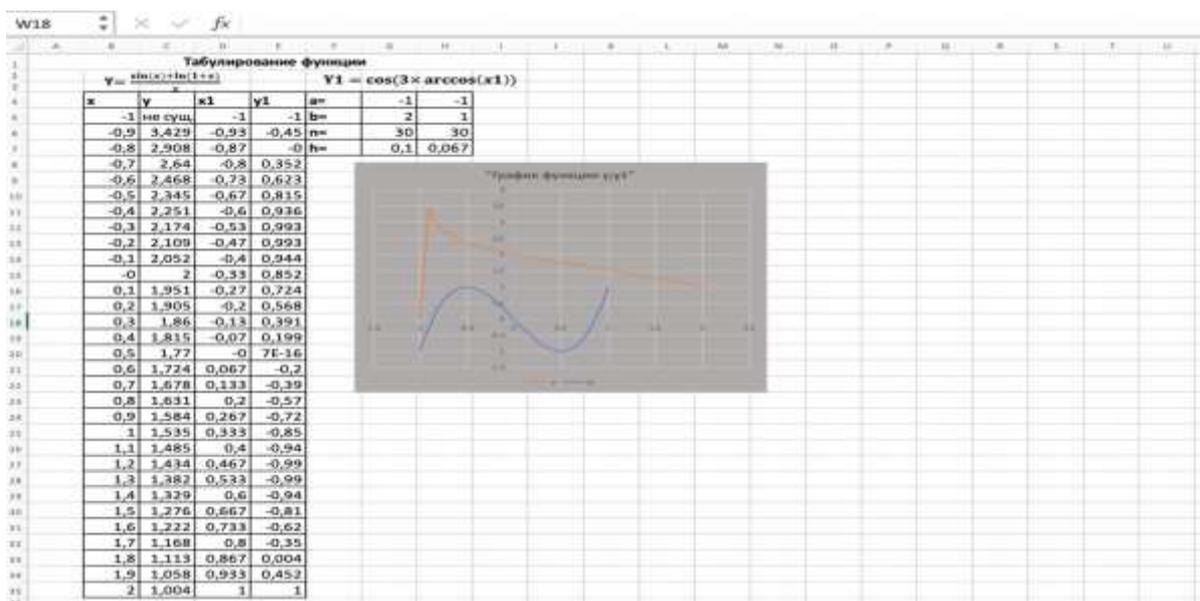


Рисунок 24 – Пример выполнения работы «Табулирование функции одной переменной»

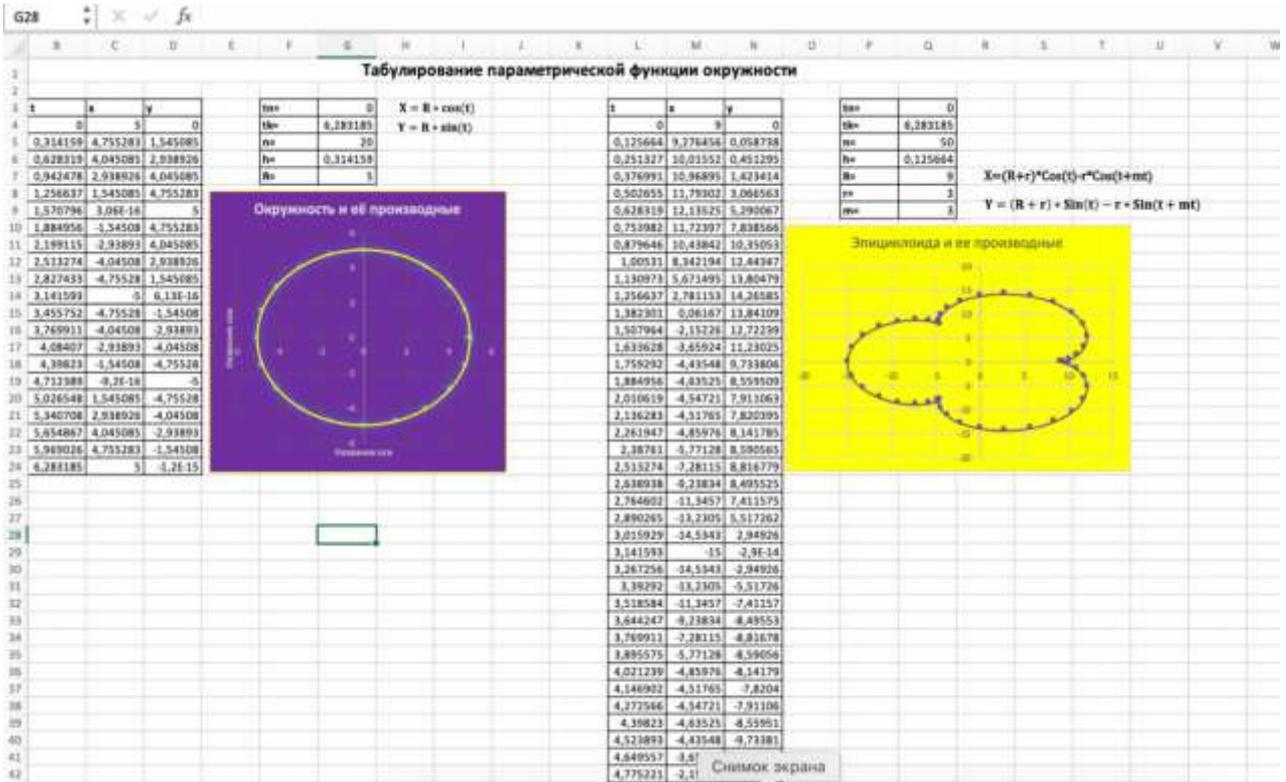


Рисунок 25 – Пример выполнения работы «Табулирование параметрической функции»

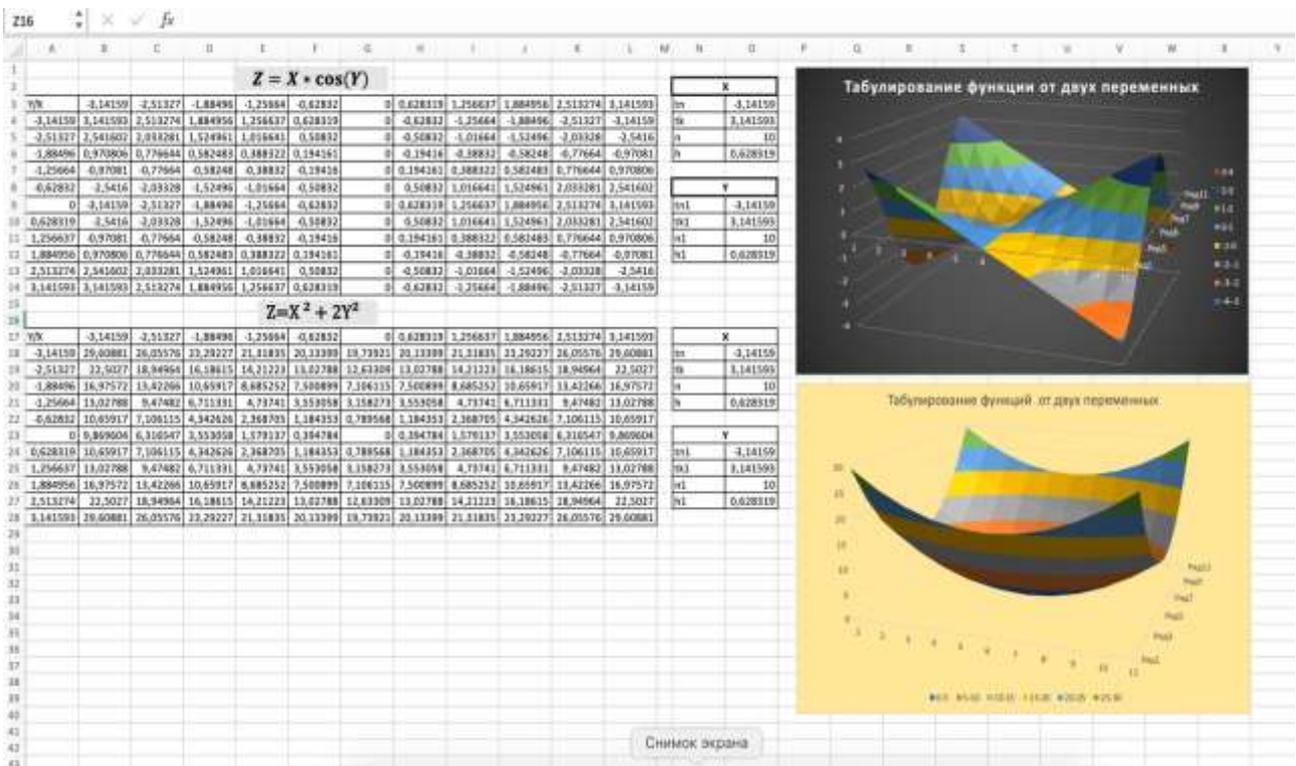


Рисунок 26 – Пример выполнения работы «Табулирование функции двух переменных»

## 2. Создание и использование базы данных MS Access

### 2.1. Логическое и физическое проектирование базы данных MS Access

Цель работы: научиться анализировать предметную область (ПО) информационной системы и на основе анализа создавать логическую и физическую модели данных в среде СУБД MS Access.

#### Основные понятия

Автоматизированная информационная система (АИС) – это комплекс программных и технических средств, обеспечивающих сбор, обработку и манипулирование данными.

Цель любой информационной системы (ИС) – обработка данных об объектах реального мира. Основой ИС является *база данных* (БД).

В широком смысле слова БД – это совокупность сведений о конкретных объектах реального мира в какой-либо ПО.

В узком смысле БД – это поименованная, определенным образом организованная совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой ПО.

Под ПО принято понимать часть реального мира, подлежащего изучению для организации управления.

Объектом называют элемент ПО, информация о котором интересует пользователя. Каждый объект описывается совокупностью свойств – *атрибутов*.

Атрибутом называют поименованную характеристику объекта. Он показывает, какая информация об объекте интересует пользователя и должна храниться в БД. Процесс проектирования БД состоит из двух этапов: *логического* и *физического* проектирования.

Сущность (информационный объект) (ИО) – любой конкретный (реальный) или абстрактный объект в рассматриваемой ПО.

Связь – наблюдаемая взаимосвязь (ассоциация) между сущностями.

Типы связей между сущностями: один – к одному ( $1 \leftrightarrow 1$ ), один – ко многим ( $1 \leftrightarrow \infty$ ), многие – к одному ( $\infty \leftrightarrow 1$ ), многие – ко многим ( $\infty \leftrightarrow \infty$ ).

Связь  $1 \leftrightarrow 1$ : в любой момент времени каждому экземпляру первого ИО соответствует 1 или 0 экземпляров другого ИО и наоборот.

Связь  $1 \leftrightarrow \infty$ : одному экземпляру первого ИО соответствует 0,1,2,... экземпляров другого и наоборот, каждому экземпляру второго ИО соответствует 0 или 1 экземпляр первого ИО. Аналогично определяется тип связи  $\infty \leftrightarrow 1$ .

Связь  $\infty \leftrightarrow \infty$ : одному экземпляру первого ИО соответствует 0,1,2,... экземпляров другого ИО и наоборот.

### **Этап логического проектирования**

Основной задачей логического проектирования является разработка логической схемы (модели), ориентированной на выбранную систему управления базами данных (СУБД). СУБД – это комплекс программных и языковых средств, предназначенных для создания, ведения и совместного применения БД любыми пользователями. СУБД осуществляет централизованное управление базой данных, обеспечивает доступ к данным, по сути, выступает в качестве интерфейса между пользователями и БД. Одним из основных критериев выбора СУБД является оценка того, насколько эффективно внутренняя модель данных, поддерживаемая системой, способна описать концептуальную схему ПО. Существующие СУБД делятся по типам моделей данных на *реляционные, иерархические* и *сетевые*.

Модель данных (МД) – формально определенная структура, которая используется для представления данных. Иерархическая МД организует данные в виде древовидной структуры, сетевая – в виде сетевой, реляционная МД – в виде таблиц (отношений). Процесс логического проектирования состоит из следующих действий:

- выбор конкретной СУБД;
- отображение концептуальной схемы на логическую схему, получение логической МД, соответствующей внешнему уровню архитектуры любой автоматизированной ИС;
- выбор ключей;
- описание языка запросов.

Теория конструирования реляционных БД разработана доктором Е. Ф. Коддом в 1968 г. Сформулированные им 13 правил полностью реализованы в реляционной СУБД MS Access. Им же введены основные понятия реляционной БД: тип данных, домен, атрибут, кортеж, первичный ключ, отношение. Понятие «тип данных» в реляционной МД полностью идентично понятию «тип данных» в языках программирования.

## Фундаментальные (базовые) свойства отношений

Домен определяется как допустимое потенциальное множество значений данного (стандартного) типа.

Кортеж, соответствующий данной схеме отношения, – это множество пар {имя атрибута, значение}, которое содержит одно вхождение каждого имени атрибута, принадлежащего схеме отношения, а «значение» – допустимое значение домена данного атрибута.

Отношение – это множество кортежей, соответствующих одной схеме.

Отсутствие кортежей-дубликатов. Из этого свойства вытекает наличие у каждого отношения первичного ключа (ПК) – набора атрибутов, значения которых однозначно определяют кортеж отношения. Понятие ПК является исключительно важным в контексте с понятием целостности БД.

Целостность базы данных – свойство, определяемое способностью СУБД защищать компоненты и связи БД от искажений в результате некорректных операций или сбоев и отказов технических средств.

Атомарность значений атрибутов. Значения всех атрибутов являются атомарными. Это следует из определения домена как потенциального множества значений одного простого типа данных, т.е. среди значений домена не могут содержаться множества (т.е. одна ячейка таблицы – одно значение).

Реляционная модель данных (РМД) – совокупность связанных таблиц. Реляционную модель можно условно разделить на три части, описывающие разные аспекты реляционного подхода: *структурную, манипуляционную и целостную*.

В целостной части РМД фиксируются два базовых требования целостности, которые должны поддерживаться в любой реляционной СУБД: *целостность сущностей* и *целостность по ссылкам*.

Требование *целостности сущностей* означает, что любой кортеж любого отношения отличим от любого другого кортежа *этого* отношения, другими словами, любое отношение должно иметь *первичный ключ*. Это требование автоматически удовлетворяется, если в системе не нарушаются базовые свойства отношений.

Требование целостности по ссылкам иначе называют требованием *внешнего ключа*.

Напомним, первичный ключ отношения (таблицы) – один или несколько атрибутов (полей), значения которых однозначно опре-

деляют кортеж (запись) таблицы. Связь между отношениями (таблицами) осуществляется путем включения общих полей. Поле *Первичного ключа* одной (говорят, главной) таблицы присутствует в качестве обычного поля в связанной (подчиненной) таблице, его и называют внешним ключом по отношению к главной таблице. Требование целостности по ссылкам означает, что для каждого значения *внешнего ключа* подчиненного отношения должен быть кортеж с таким же значением *первичного ключа* главного отношения либо значение *внешнего ключа* должно быть неопределенным (ни на что не указывать). Ограничения целостности должны поддерживаться СУБД. Для соблюдения целостности сущностей достаточно гарантировать отсутствие в любом отношении кортежей с одним и тем же значением первичного ключа. В обеспечении целостности по ссылкам существуют три подхода.

Первый подход заключается в том, что *запрещается* производить *удаление* кортежа, на который существуют ссылки, т.е. сначала нужно либо удалить ссылающиеся кортежи, либо изменить значения из внешнего ключа должным образом.

При втором подходе при *удалении* кортежа, на который имеются ссылки, во всех ссылающихся кортежах значение внешнего ключа автоматически становится неопределенным.

Третий подход (каскадное удаление) состоит в том, что при *удалении* кортежа из отношения, на которое есть ссылка, автоматически удаляются все ссылающиеся кортежи. В современных СУБД (в том числе MS Access) обычно можно выбрать способ поддержания целостности по ссылкам для каждой отдельной ситуации (связи) определения внешнего ключа.

## **Нормализация БД**

При проектировании реляционных БД большое внимание уделяется нормализации таблиц. В процессе нормализации обеспечивается защита целостности данных путем устранения их дублирования. В результате исходная таблица разбивается на две или более связанных таблиц, которые могут быть «собраны» вместе с помощью операции объединения. Руководство по нормализации – это набор стандартов (правил) проектирования данных, называемых нормальными формами (НФ).

Общепринятыми считаются пять нормальных форм, хотя их было предложено больше. Создание таблиц в соответствии с этими стандартами называется нормализацией.

Четвертая НФ запрещает независимые отношения типа «один – ко многим» между ключевыми и неключевыми столбцами.

Ориентируясь на СУБД MS Access, которая поддерживает реляционную модель данных, разработаем логическую модель данных предметной области предприятия «Сервис-центр» (рис. 27).

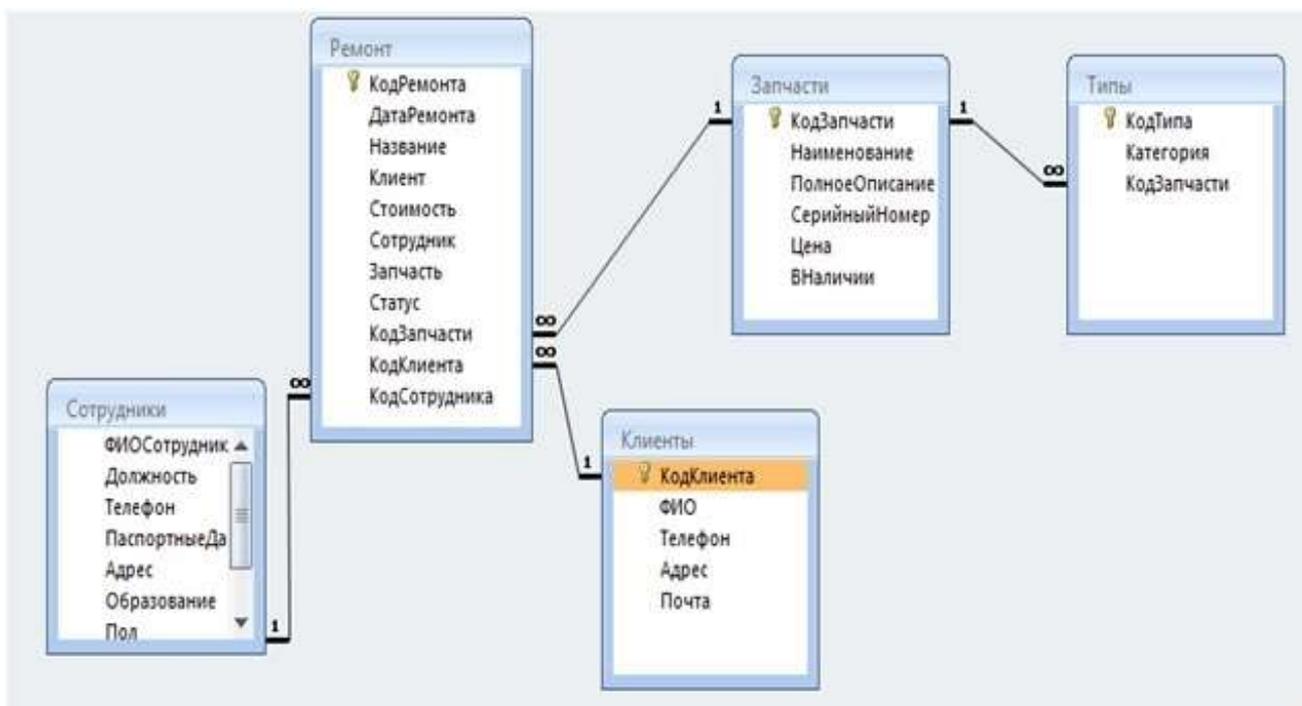


Рисунок 27 – Пример физической модели БД

Все таблицы имеют четвертую НФ. В таблице Ремонт простой первичный ключ – поле *Код ремонта*. В таблице Запчасти простой первичный ключ – поле *Код запчасти*. В таблице Сотрудник простой первичный ключ – поле *Код сотрудника*.

Следующий этап – физическое проектирование: логическая модель данных отображается в физическую, в результате получается схема данных, определяющая размещение данных, методы доступа и технику индексирования. Физическая модель соответствует внутреннему уровню архитектуры любой АИС. В современных СУБД (в том числе MS Access) процесс физического проектирования БД осуществляется автоматизированно средствами самой СУБД.

## Этап физического проектирования

Итак, к этому моменту в качестве среды реализации *базы данных* (БД) выбрана СУБД *MS Access*. Логическая модель данных (МД) отображается в физическую модель (рис. 27), которая определяет размещение и наполнение БД в памяти компьютера средствами СУБД.

СУБД *MS Access* – реляционная система управления базами данных, поэтому *БД MS Access* – это совокупность взаимосвязанных реляционных таблиц (фундаментальные свойства отношений выполнены), в каждой из которых содержатся сведения об одной сущности. Каждый столбец таблицы называют *полем*, строку – *записью*.

Понятия *первичного* и *внешнего* ключей общей теории проектирования реляционных БД совпадают с аналогичными понятиями в *MS Access*.

В СУБД *Access* процесс физического проектирования БД включает создание таблиц (определение структуры) и создание связей между ними с помощью *схемы данных*. *Схема данных* – графическое представление физической МД – полный аналог логической модели.

*MS Access* ориентирована на работу с различными объектами, которые включают таблицы БД, запросы и объекты приложений для манипулирования данными: *формы, отчеты, страницы, макросы и модули*. Все объекты БД, как правило, хранятся в одном файле формата \*.accdb. Страницы доступа к данным *MS Access* хранит в отдельных файлах с расширением \*.htm, в файле БД сохраняются только ссылки на них.

## Технология создания БД в MS Access

Для запуска *MS Access* нажать кнопку **Пуск**, выбрать **Программы, MS Access** или на рабочем столе операционной системы Windows щелкнуть мышью ярлык приложения *MS Access* (или другим способом). Открывается *Диалоговое окно* (ДО) приложения, которое имеет все элементы стандартного графического интерфейса приложений, работающих под управлением ОС Windows (*строка меню, панели инструментов* и т.д.).

Одновременно с окном приложения выводится первое ДО, позволяющее создать новую БД или открыть существующую (рис. 28).

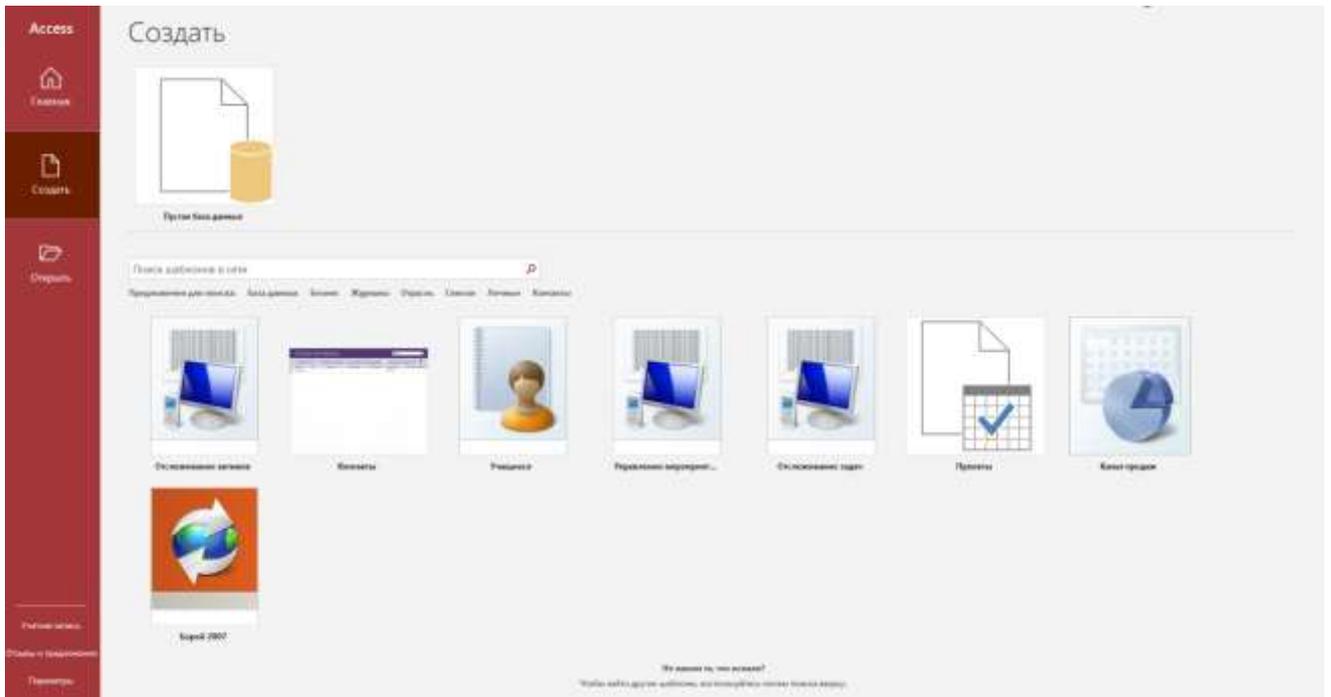


Рисунок 28 – Создание базы данных

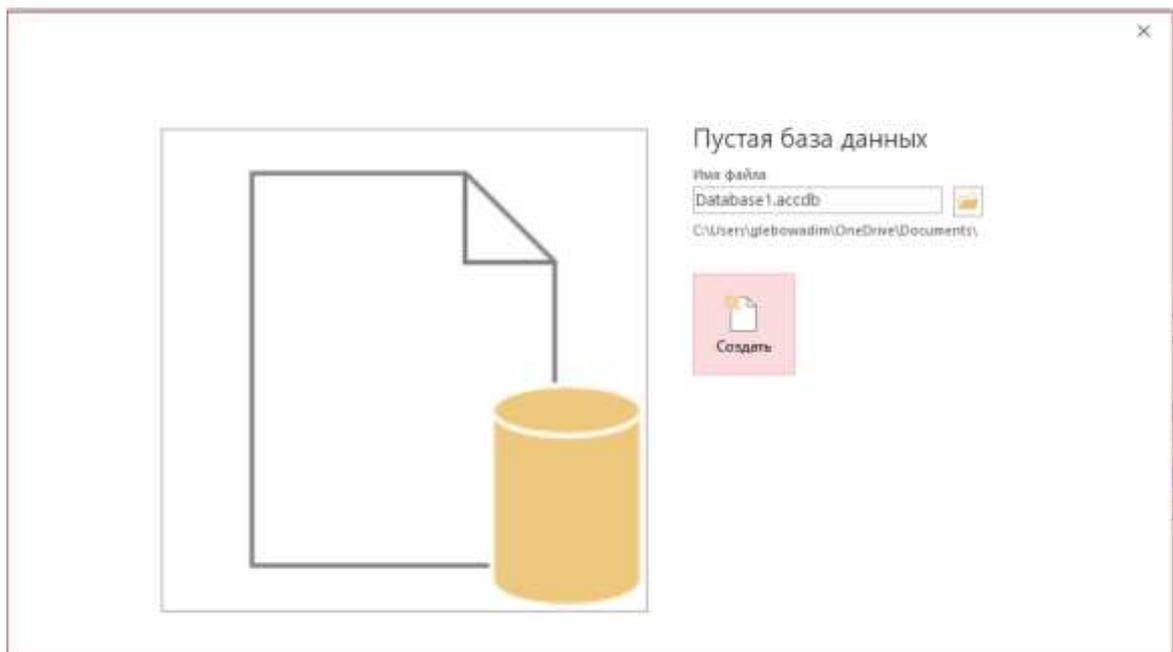


Рисунок 29 – Создание пустой базы данных

Следует выбрать способ создания БД (пустую или с помощью шаблона) или открыть существующую (рис. 29). Если выбрать *Новая БД*, надо справа выбрать папку для хранения БД (**z:\**) и в поле *Имя* ввести свою фамилию (или содержательное имя), нажать кнопку **Создать**, открывается ДО (рис. 30). Заголовок окна представлен на рисунке 31. Автоматически открывается макет таблицы для создания ее в режиме *таблицы*.

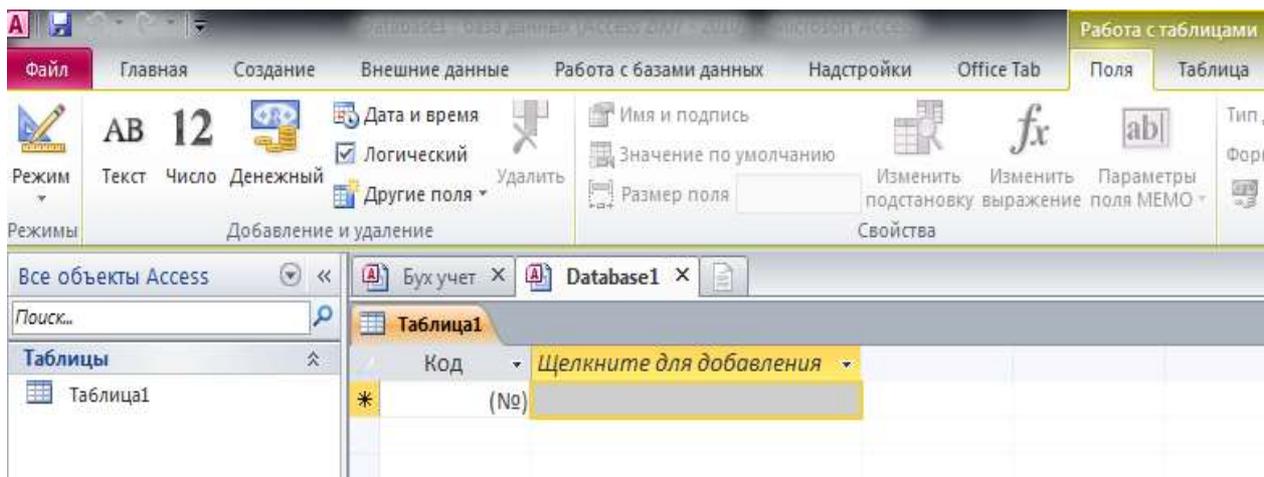


Рисунок 30 – Работа с таблицами

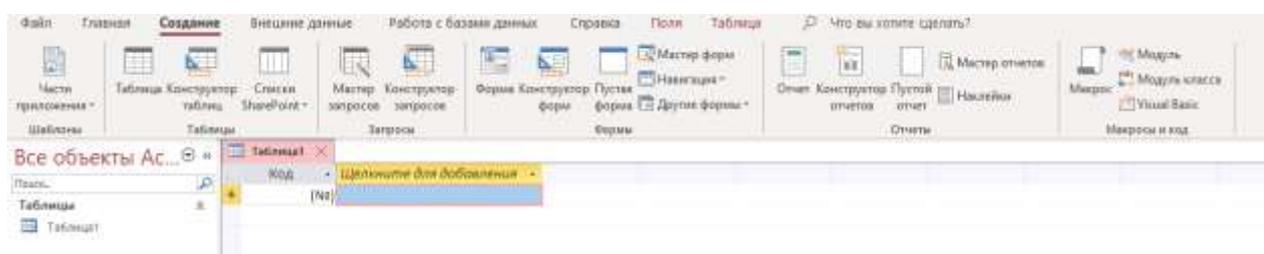


Рисунок 31 – Работа с таблицами в режиме редактирования

Слева присутствует панель **Объектов**, с помощью которой реализован доступ к существующим и вновь создаваемым объектам. Вверху слева в панели инструментов есть список *Режимы* (*режим таблицы, сводная диаграмма, сводная таблица, конструктор*). Список *Режимы* позволяет открывать различное представление объектов базы данных.

Рисунок 32 содержит шаблоны типовых БД для предметных областей различных сфер деловой и личной жизни, которые включают все необходимые таблицы, формы, запросы и отчеты. Мастер баз данных создает на основе выбранного шаблона пользовательскую БД в процессе диалога. Базу данных, созданную мастером, можно изменить и расширить, но эта работа требует от пользователя практически тех же знаний, что и создание новой БД. Строка главного меню содержит различные вкладки. Для создания объектов БД предназначена вкладка **Создание** (рис. 32), для использования – вкладка **Работа с базами данных** (рис. 33).

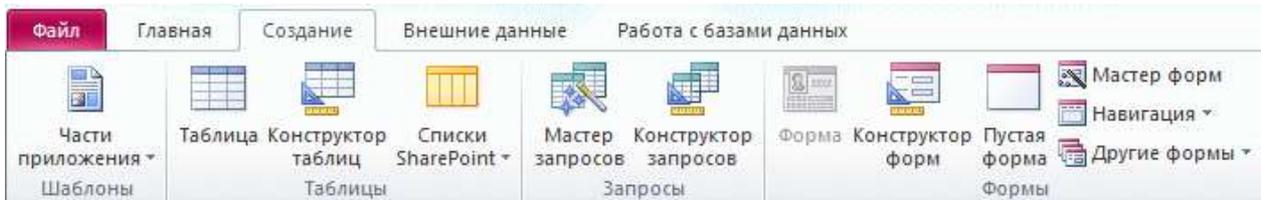


Рисунок 32 – Вкладка «Создание»

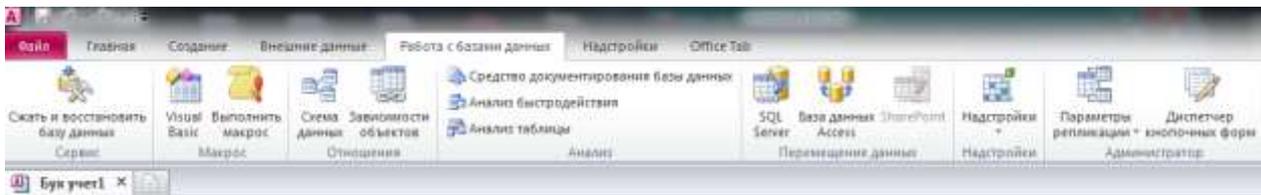


Рисунок 33 – Вкладка «Работа с базами данных»

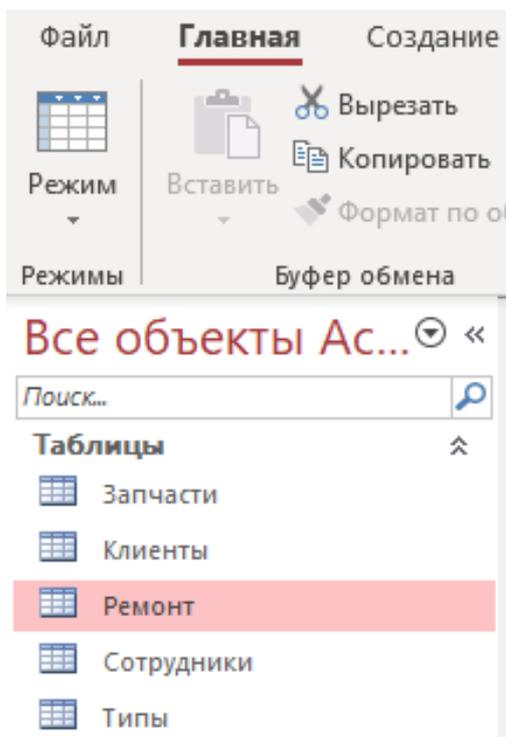


Рисунок 34 – Панель «Все объекты»

Панель *Все объекты* содержит вкладки для всех объектов, которые могут быть созданы в БД: *Таблицы, Запросы, Формы, Отчеты, Страницы, Макросы, Модули*. При выборе соответствующей вкладки в рабочем поле окна отображается список имен соответствующих созданных объектов (рис. 34).

### Создание таблиц

Создание таблицы состоит из двух этапов: определение ее структуры (состав полей, их имена, тип и размер каждого поля, ключевые поля и другие свойства) и создание записей таблицы, т.е. заполнение ее данными.

Существуют два основных способа создания новой таблицы: режим *таблица*, режим *конструктор таблиц*.

Ориентируясь на логическую модель БД «Сервис-центр» (рис. 27), создадим таблицы *Ремонт, Клиенты, Запчасти, Сотрудники* и *Типы*.

## 1 способ: режим *конструктор таблиц*

Выбираем режим *конструктор таблиц* в списке режимов (рис. 31) или на вкладке **Создание** (рис. 32), открывается окно, в котором определяется структура новой таблицы (рис. 35, рис. 36).



Рисунок 35 – Окно «Работа с таблицами»

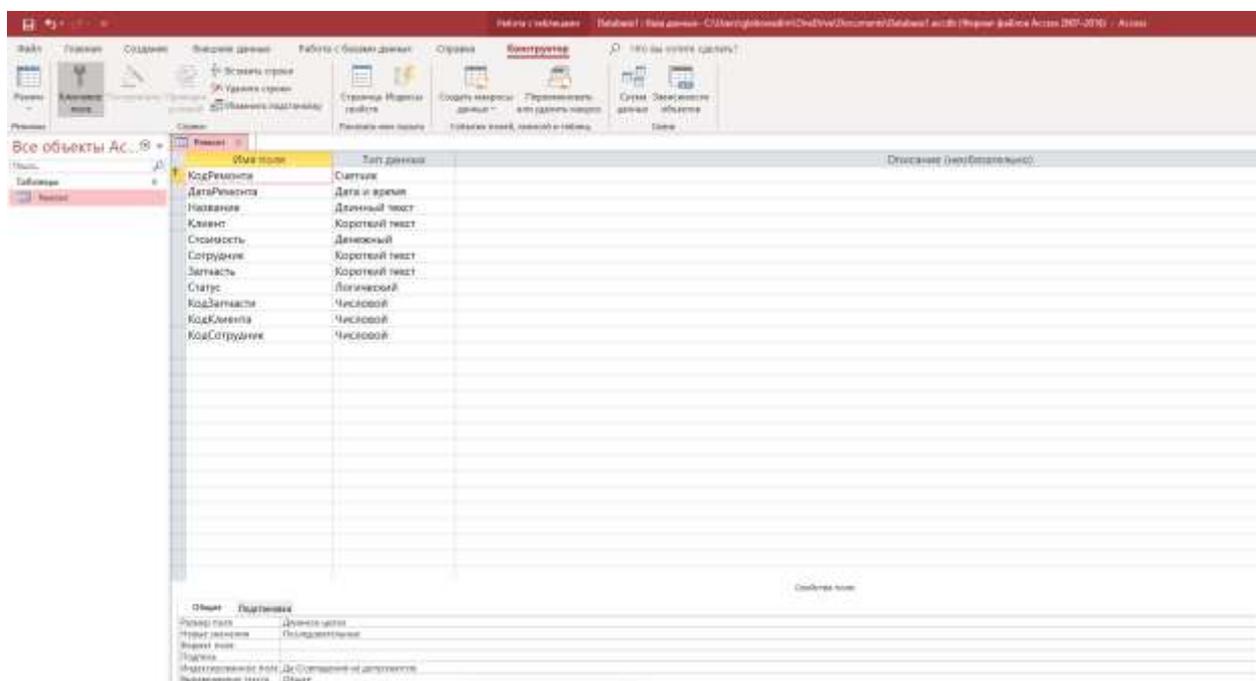


Рисунок 36 – Создание таблицы *Ремонт* в режиме *конструктора*

Окно *конструктора* состоит из двух частей: верхняя часть определяет структуру таблицы – перечень полей и их типов, нижняя часть – окно свойств выделенного поля. Переход в окно свойств и назад осуществляется мышью или клавишей **F6**.

В верхней части конструктора в первую строку столбца *Имя поля* введем идентификатор поля. Имя поля должно быть уникальным в таблице, может содержать не более 64 символов, оно должно удовлетворять соглашениям об именах объектов в Access. Имя не может начинаться с пробела и содержать управляющие символы,

нельзя использовать в качестве имен зарезервированные слова MS Access. Надо избегать длинных имен. Затем выбираем в списке *Тип данных* – *Счетчик* (или *Числовой*). Тип данных определяется значениями, которые предполагается вводить в поле.

Список возможных типов данных содержит:

- *текстовый* – строка до 255 символов;
- *поле МЕМО* – тоже поле текстового типа, может содержать до 65535 символов, предназначено для хранения комментариев, не может быть ключевым полем; ввод данных в это поле можно осуществить в режиме *таблицы* непосредственно в поле либо через область ввода, которая открывается при нажатии клавиши <Shift+F2>;
- *числовой* – хранит числовую информацию, длина поля зависит от значения свойства *Размер поля*;
- *дата/время* – хранит значение времени и даты, с помощью свойства *Формат поля* задается способ представления значения;
- *денежный* – используется для проведения денежных расчетов с точностью до 15 знаков в целой и 4 знаков в дробной частях;
- *счетчик* – по умолчанию значения поля уникальные целые числа, последовательно возрастающие на единицу; значение этого поля нельзя изменить или удалить (ввести); используется в качестве первичного ключа таблицы;
- *логический* – хранит логические данные, имеющие одно из двух возможных значений: да/нет, истина/ложь, вкл./выкл.;
- *поле объекта OLE* – используется для вставки объекта другого приложения, например, рисунка, фотографии, звука и т.д. (меню *Вставка, Объект*);
- *гиперссылка* – можно указать путь к файлу на жестком диске или адрес (универсальный указатель ресурса) Web-страницы в сети Интернет;
- *мастер подстановки* – выбор этого типа данных запускает *Мастер подстановок*, обычно используется для внешних ключей, реализует первый способ создания поля подстановки (поле с раскрывающимся списком) в режиме *конструктора таблиц*.

### **Свойства полей**

Размер поля – задает максимальный размер данных, хранимых в поле в соответствии с выбранным типом данных. Например, для поля *№ операции* таблицы *ЖХО*, имеющего тип *Счетчик*, выберем из списка *Длинное целое*. Рекомендуется задавать минимально допустимый размер поля.

Формат поля – задает вид отображения значений полей всех типов, кроме *Поля объекта OLE*. В Access определены в виде раскрывающихся списков встроенные стандартные форматы для полей типов: *Числовой*, *Дата/Время*, *Логический* и *Денежный*. Пользователь может создать собственный формат с помощью символов форматирования.

Маска ввода (МВ) – свойство предназначено для облегчения ввода и контроля вводимой информации. МВ строится с помощью мастера, выбирается из списка (для некоторых типов полей) или самостоятельно пользователем на основе таблицы специальных символов, которую можно увидеть в справочной системе MS Access.

Оба свойства *Формат поля* и *Маска ввода* задают шаблон, в котором данные вводятся в таблицу. Свойство *Формат поля* используется для данных *числового* и *денежного* типа и не определено для текстовых полей, полей МЕМО и гиперссылок. Свойство *Маска ввода* используется для отображения символьных констант и при вводе данных в пустые графы. Маска обеспечивает сохранение информации в определенном формате. Обычно задано одно из свойств. Если заданы одновременно оба свойства, то свойство *Маска ввода* применяется при вводе и редактировании данных, а *Формат поля* – при отображении сохраненных данных. Если щелкнуть кнопку **Построителя**  у свойства МВ, запустится *Мастер маски ввода*. Лучше задавать вручную, используя таблицу кодов.

Главное отличие свойств: *Формат поля* влияет на все содержимое поля, а *МВ* – на часть.

Подпись – свойство задает заголовок столбца при выводе в таблицах, формах, отчетах; если значение не задано, используется имя поля.

Значение по умолчанию – свойство, присущее полям всех типов, за исключением полей типа *Счетчик* и *Объект OLE*. Значение этого свойства автоматически подставляется при создании новой записи. Можно использовать выражение, например, **=Date()** – вывод по умолчанию текущей даты, знак «равно» обязателен.

Условие на значение и Сообщение об ошибке – эти свойства применимы ко всем типам полей, кроме *Счетчик* и *Объект OLE*. При переходе на другую запись проверяется указанное условие; если оно не выполняется, выдается сообщение об ошибке – значение второго свойства. Например, условие **<>#01.01.2002# and**

<#01.01.2003#> задает диапазон введенной даты. Константа типа Date обрамляется с обеих сторон знаком диеза «#».

Обязательное поле – свойство можно указать для всех полей, кроме *Счетчика* (оно задано по умолчанию). Возможные значения «Да», «Нет» выбираются из списка. Если указано «Да», то ввод значения поля обязателен, иначе появляется сообщение об ошибке.

Пустые строки – разрешает или запрещает ввод пустых строк, возможны два значения: «Да», «Нет». Значение «Да» обычно используется для текстовых полей, *МЕМО* и *гиперссылок*.

Индексированное поле – указывает, будет ли создан индекс по этому полю. Возможные значения: «Нет», «Да (Допускаются совпадения)», «Да (Совпадения не допускаются)». *Индекс* – вспомогательная таблица, в которой хранится порядок следования записей, упорядоченных по этому полю. Для первичного ключа таблицы указываем – «Да (Совпадения не допускаются)».

Access автоматически индексирует таблицу по значению ключа. При создании запросов и отчетов может потребоваться создать дополнительные индексы по значениям других полей. Индексы позволяют ускорить поиск данных в больших таблицах. Каждая таблица может иметь до 32 индексов, 5 из которых могут быть составными, в составной индекс могут входить не более 10 полей. Чтобы создать дополнительный *простой ключ*, надо в режиме *конструктора* задать свойство *Индексированное поле* – «Да (Совпадения допускаются)». Чтобы создать дополнительный *составной ключ*, надо в режиме *конструктора* открыть окно *Индексы*, нажав соответствующую кнопку **Индексы** на ПИ. В этом окне (рис. 37) выводятся все индексы таблицы. Надо в поле *Индекс* внести имя индекса (произвольное), в поле *Имя поля* выбрать из списка сначала одно поле, перейти на строку ниже, выбрать другое и т.д. Закрывать окно, подтвердить сохранение структуры таблицы.

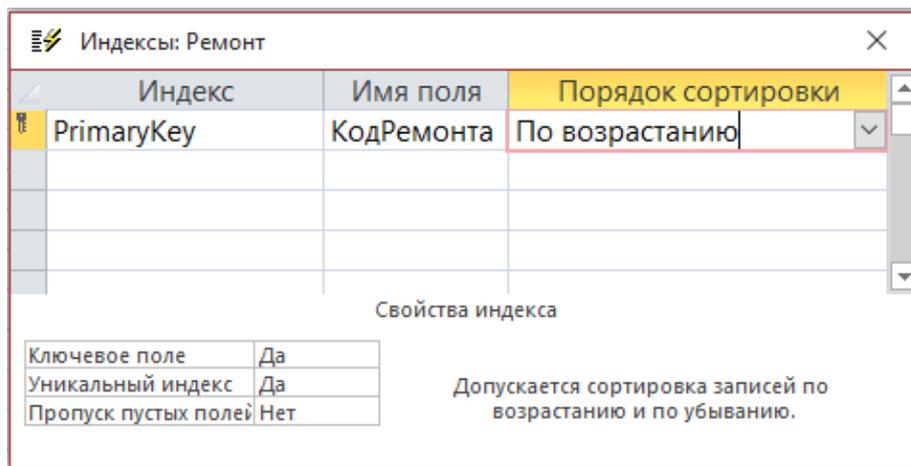


Рисунок 37 – ДО Индексы

Сжатие Юникод – включает и отключает сжатие *текстовых полей, полей МЕМО и гиперссылок*, используя универсальный международный код. По умолчанию включено. Можно отключить, тем самым сэкономить память.

Число десятичных знаков – указывает количество десятичных знаков после запятой в *числовых и денежных* типах полей; если выбрать значения свойства *Авто*, то число отображается так, как было введено в поле.

Новые значения – свойство определено только для поля *Счетчик* и указывает, какие значения автоматически будут подставляться в поле при создании новой записи. Возможные значения: *последовательные, случайные* (числа). Принято по умолчанию первое значение и лучше использовать его.

Определим другие свойства полей. Переходим в верхнюю часть конструктора, выделим поле, щелкнув мышью слева от поля в области стрелки (если ключ простой, можно не выделять поле, но оно должно быть выбрано), и нажимаем кнопку  **Ключевое поле** на ПИ, значок ключа появляется слева от имени поля (рис. 37), автоматически создается соответствующий индекс таблицы.

Аналогично создадим структуру таблиц *Клиенты, Сотрудники, Запчасти, Типы* (рис. 38, 39, 40, 41).

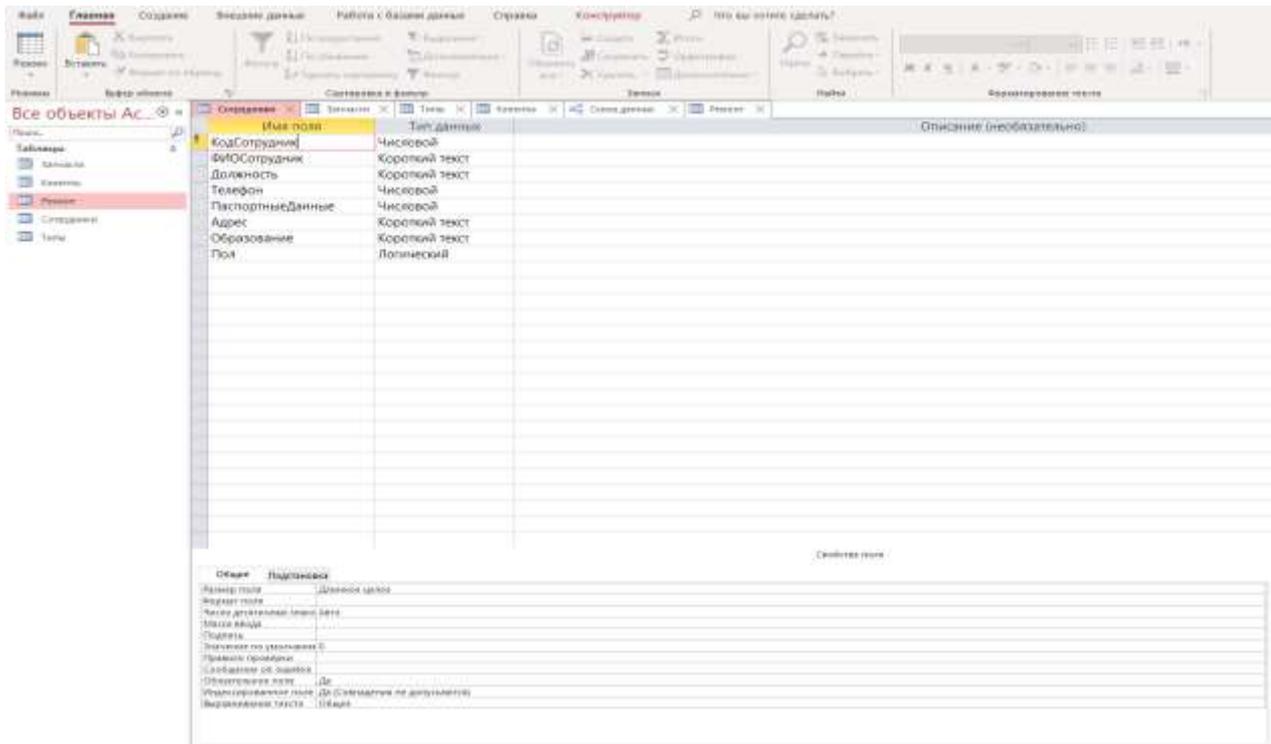


Рисунок 38 – Окно конструктора таблицы *Сотрудники*

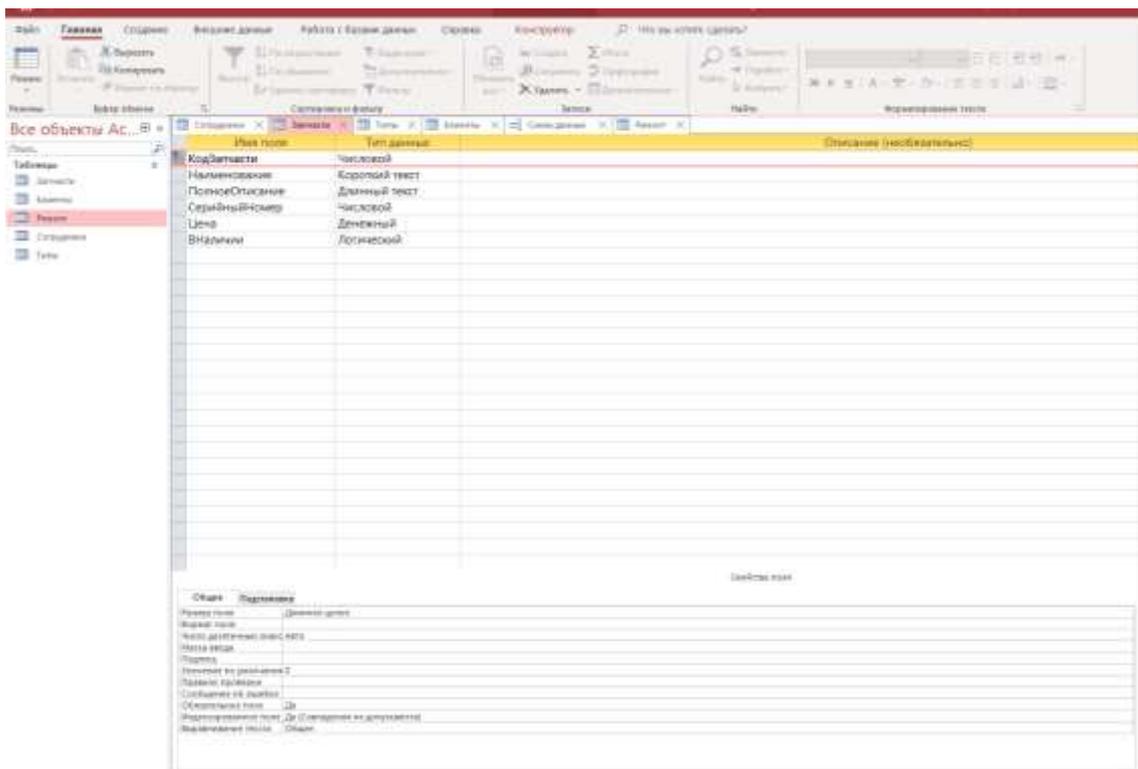


Рисунок 39 – Окно конструктора таблицы *Запчасти*

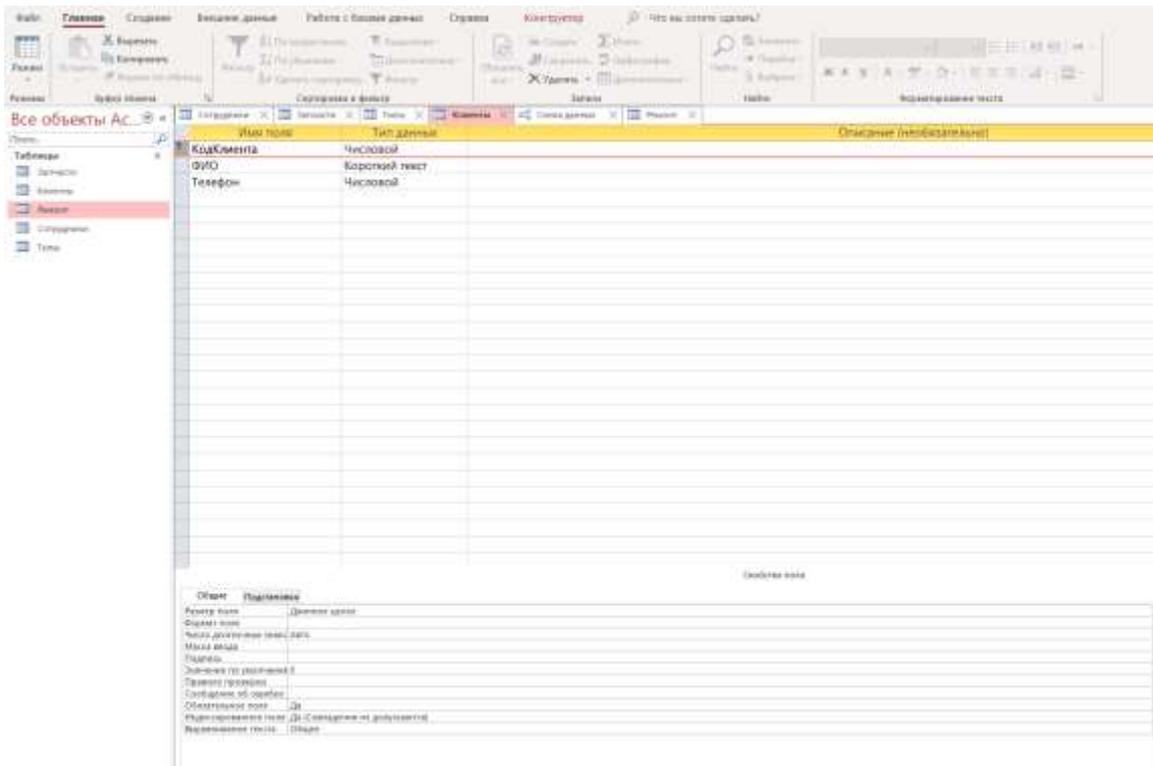


Рисунок 40 – Окно конструктора таблицы *Клиенты*

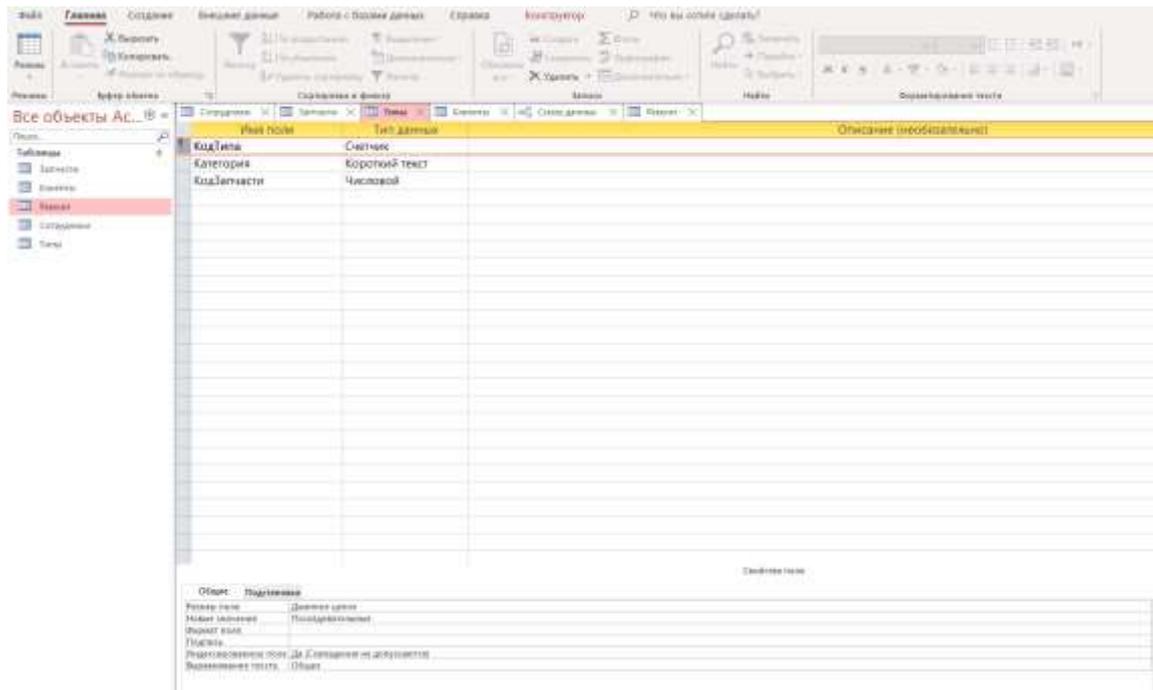


Рисунок 41 – Окно конструктора таблицы *Типы*

Сразу после сохранения структуры таблицы можно заполнить ее данными (создать записи), для этого на вкладке **Таблицы** выбрать нужную. Лучше наполнить таблицу данными позже через формы, иначе можно нарушить целостность БД.

## 2 способ: режим *таблица*

В режиме *таблицы* создается новая таблица без предварительного определения ее структуры. Введенные данные автоматически определяют тип поля таблицы, т.е. автоматически создается структура таблицы. Имя поля можно изменить, дважды щелкнув левой кнопкой мыши по имени в области заголовков или используя команду меню *Формат, Переименовать столбец*. Можно вставить (команда меню *Вставка, Столбец*) или удалить (команда меню *Правка, Удалить столбец*) столбцы.

Для перемещения столбца надо выделить имя и мышью перетащить заголовок поля в нужное место. Все столбцы и строки, оставленные пустыми, будут удалены при сохранении таблицы. Действующие форматы полей можно увидеть в окне конструктора таблиц, свойство *Формат поля*. По окончании ввода данных нужно сохранить таблицу. При сохранении таблицы появится сообщение: «Создать ключевое поле? Да, Нет». Надо выбрать «Нет», затем открыть таблицу в режиме *конструктора*, изменить при необходимости структуру и свойства полей, задать ключи.

В режиме *таблицы* надо вводить данные правильно в соответствии с допустимыми типами данных и форматами их отображения. В окне конструктора таблиц можно просмотреть их определение.

## Использование полей подстановок при создании таблиц

В связанных таблицах внешний ключ подчиненной таблицы должен иметь значение первичного ключа главной таблицы, иначе нарушаются параметры целостности БД.

## Подстановка в режиме *конструктора таблиц*

Первый способ. Переходим в окно *свойств*. Вкладка **Общие** – свойство *Размер поля* – 6, открываем вкладку **Подстановка**, определим свойства: *Тип элемента управления* – поле со списком, *Источник строк* – таблица *Запчасти*, *Присоединенный столбец* – 1, *Число столбцов* – 1 (рис. 42).

Общие	Подстановка
Тип элемента управления	Поле со списком
Тип источника строк	Таблица или запрос
Источник строк	SELECT [Запчасти].[КодЗапчасти] FROM Запчасти;
Присоединенный столбец	1
Число столбцов	1
Заголовки столбцов	Нет
Ширина столбцов	1"
Число строк списка	16
Ширина списка	1"
Ограничиться списком	Нет
Разрешить несколько значений	Нет
Разрешить изменение списка	Нет
Форма изменения элемента	
Только значения источника	Нет

Рисунок 42 – Определение поля подстановки

Второй способ. Выбрать поле, в свойстве *Тип данных* указать: *Мастер подстановок*. Появится ДО, в котором надо указать вариант.

• *Объект «столбец подстановки» будет использовать значения из таблицы или запроса или*

• *Будет введен фиксированный набор значений.* Нажать кнопку **Далее**.

Если выбран первый вариант, появится следующее ДО, в котором надо указать • *Таблицы* и в списке выделить таблицу *Запчасти*, нажать кнопку **Далее**. В следующем ДО выбрать поле *Код запчасти* кнопкой , которое определит столбец подстановки, кнопка **Далее**; в следующем ДО появится столбец значений, в этом ДО можно изменить ширину столбца, кнопка **Готово**. Можно ввести фиксированный набор значений.

### Установка связей между таблицами

Создание связей между таблицами – последний этап физического проектирования БД. Связь между таблицами устанавливает отношение между совпадающими значениями в ключевых полях. В большинстве случаев связывают первичный ключ главной таблицы с внешним ключом подчиненной таблицы (часто имеющим то же имя), например, первый ключ *Код контрагента* таблицы *Контрагенты* и внешний ключ *Код контрагента* таблицы *ЖХО*. Существуют четыре типа отношений:

1. Один – к одному 1 ↔ 1. Запись таблицы *A* может иметь не более одной связанной записи в таблице *B* и наоборот. Ключевое поле в таких таблицах должно быть уникальным.

2. Многие – к одному  $\infty \leftrightarrow 1$ . Одной записи в таблице *A* может соответствовать одна запись в таблице *B*, а одной записи в таблице *B* – несколько записей в таблице *A*. В таблице *B* ключевое поле должно быть уникальным.

3. Один – ко многим  $1 \leftrightarrow \infty$ . Каждой записи в таблице *A* могут соответствовать несколько записей в таблице *B*, а запись в таблице *B* не может иметь более одной соответствующей записи в таблице *A*. В таблице *A* ключевое поле должно быть уникальным. Отношения вида  $1 \leftrightarrow \infty$  и  $\infty \leftrightarrow 1$  отличаются тем, какая таблица является главной.

4. Многие – ко многим  $\infty \leftrightarrow \infty$ . Такая таблица не нормализована. Одной записи таблицы *A* может соответствовать несколько записей таблицы *B* и наоборот. Уникальных ключей нет. Все ключи внешние. В этом случае таблицу надо нормализовать, используя инструмент *Анализатор таблиц* (меню *Сервис, Анализ, Таблица*, запускается мастер).

Связанные поля не обязательно должны иметь одинаковые имена, но обязательно должны иметь одинаковые типы данных. Поля типа *Числовой* должны иметь одинаковые значения свойства *Размер поля*. Разрешается связывать типы полей *Счетчик* и *Числовое*, *Длинное целое* (или *Счетчик* и *Числовое поле*, если код репликации задан для обоих полей). Перед установкой связей следует закрыть все таблицы.

На вкладке **Работа с базами данных** нажать кнопку **Схема данных**, затем **Добавить таблицу** (или команда меню *Связи, Добавить таблицу*), появится окно *Добавление таблиц*. Выбрать таблицу из списка (один раз щелкнуть левой кнопкой мыши) и нажать кнопку **Добавить**, выбрать следующую таблицу, нажать кнопку **Добавить** и т.д. Выбрать все таблицы, затем нажать кнопку **Заккрыть**. Затем уцепить левой кнопкой мыши поле первой таблицы со стороны связи 1 и тащить, опустить на то же поле другой таблицы, появится ДО *Изменение связей* (рис. 43), внизу указан тип отношения *один – ко многим*, нажать кнопку **Создать**.

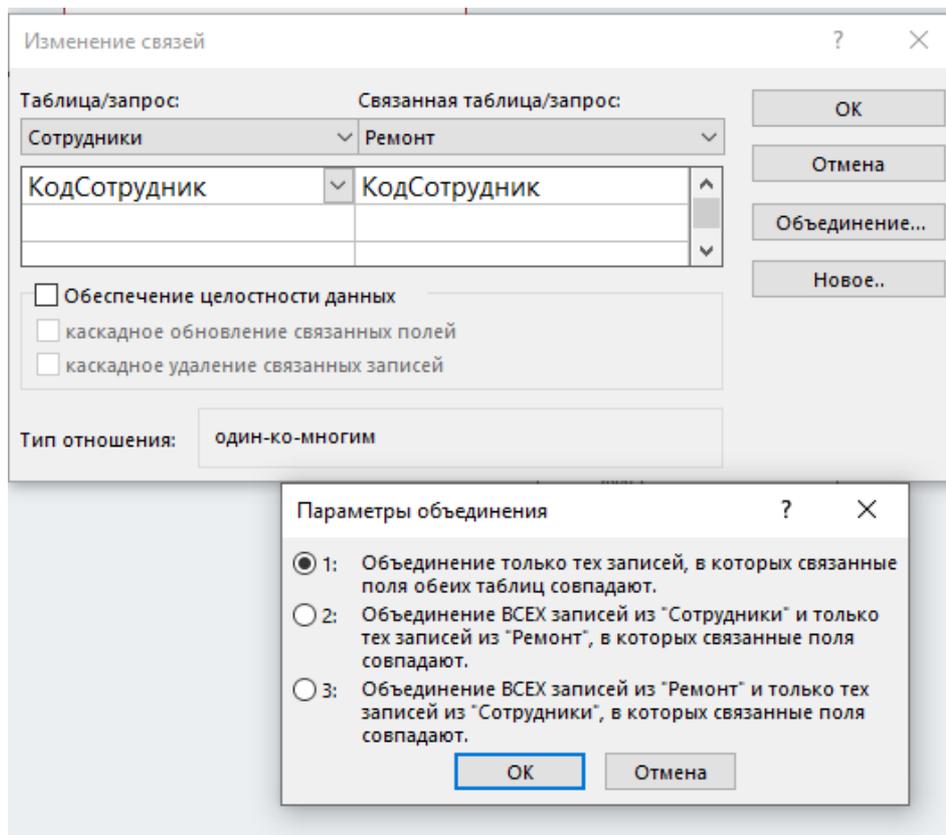


Рисунок 43 – Изменение связей. Параметры объединения

Для удаления связи нужно щелкнуть правой кнопкой мыши нужную связь и выбрать команду *Удалить*, для изменения – два раза щелкнуть линию связи, появится ДО *Изменение связей*, которое имеет три флажка:  *Обеспечение целостности данных* означает выполнение условий корректного внесения записей,  *Каскадное обновление связанных полей*,  *Каскадное удаление связанных полей* означают, что при изменении (или удалении) значений первичного ключа главной таблицы автоматически меняются (или удаляются) соответствующие значения внешнего ключа подчиненной таблицы. Связи следует создавать при пустых таблицах, иначе нельзя будет установить эти параметры целостности БД (или какой-либо из них).

При составном ключе в окне *Изменение связей* необходимо для каждого ключа главной таблицы (Таблица) выбрать соответствующее поле подчиненной таблицы (Связанная таблица). При установлении связи по составному ключу необходимо выделить все поля, входящие в ключ главной таблицы, и перетащить их на одно из полей связи в подчиненной таблице. При выделении полей удерживать клавишу **<Ctrl>**.

Установленные связи используются при создании запросов, форм и отчетов, их называют постоянными. Надо сохранить макет схемы данных, нажав кнопку **Сохранить**. Закройте окно *Схема данных*. Установим связи между всеми таблицами в соответствии с логической моделью данных.

Если в ДО *Изменение связей* нажать кнопку **Объединение**, откроется ДО *Параметры объединения* – критерии для отбора записей (рис. 19). Например, для любого типа связи можно выбрать:

- Объединение только тех записей, в которых связанные поля обеих таблиц совпадают (внутреннее объединение).

- Объединение всех записей из таблицы *A* и только тех из таблицы *B*, в которых связанные поля совпадают (левое внешнее объединение).

- Объединение всех записей из таблицы *B* и только тех из таблицы *A*, в которых связанные поля совпадают (правое внешнее объединение).

Для связей между таблицами на уровне БД лучше использовать только внутреннее объединение, а оба внешних объединения – при создании запросов.

Для связей между таблицами на уровне БД лучше использовать только внутреннее объединение, а оба внешних объединения – при создании запросов.

В MS Access существуют два способа организации связи: создание непосредственных связей (вручную) с помощью инструмента *Схема данных* и создание вложенных таблиц. Чтобы создать вложенную таблицу, надо открыть главную таблицу в режиме *просмотра*, активизировать поле первичного ключа (необязательно) и выполнить команду меню *Вставка, Подтаблица*, появится ДО, в котором выбрать подчиненную таблицу и указать: в списке *Подчиненные поля* – внешний ключ (первичный ключ подчиненной таблицы) и в списке *Основные поля* – первичный ключ главной таблицы, ОК. Рядом с именем поля первичного ключа главной таблицы появится знак **+**. При щелчке мыши по нему открывается подчиненная (связанная) таблица.

Схема данных – графическое представление логической модели БД (рис. 44). Открыть *Схему данных* можно на вкладке **Работа с базами данных**, там же можно получить Отчет по схеме данных (рис. 45).

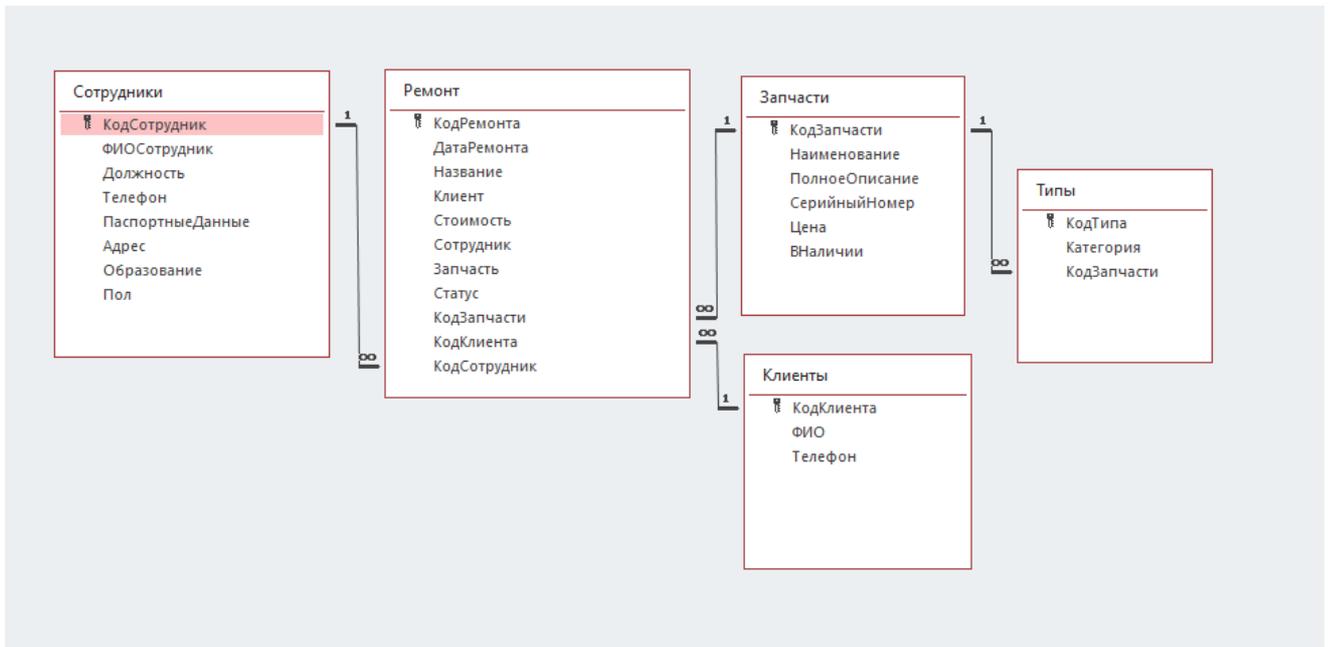


Рисунок 44 – Схема данных

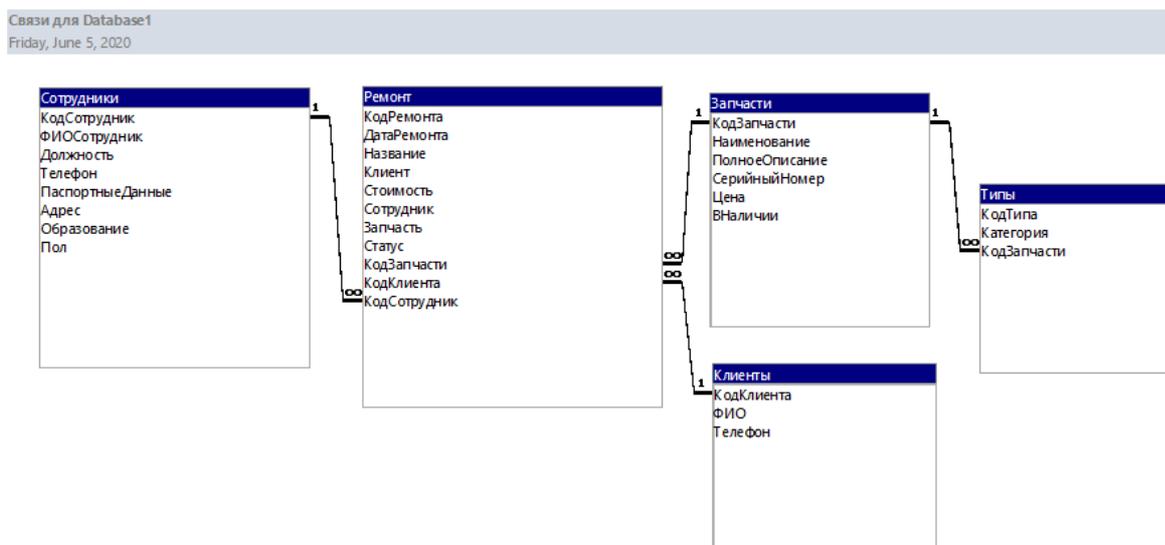


Рисунок 45 – Отчет по схеме данных

### 2.1.1. Задание по лабораторной работе

1. Спроектировать базу данных, состоящую из четырех–пяти таблиц, описывающих определенную предметную область ИС. Каждая запись таблицы должна состоять не менее чем из пяти–восьми разнотипных полей.
2. Определить ключи таблиц и типы связей между ними.
3. Используя режим *конструктора*, создать таблицы БД своей предметной области в соответствии с логической МД.
4. Определить типы данных полей, включая *MEMO*, *объект OLE* и выбор из списка (*Мастер подстановок*).

5. Для некоторых полей задать свойства *Маска ввода*.
6. Для каждой таблицы определить *первичные* и *внешние* ключи.
7. Для *внешнего* ключа подчиненной таблицы определить *поле подстановки*, указав в качестве столбца данных поле *первичного* ключа главной таблицы.
8. Создать схему данных. В диалоговом окне *Изменение связей* включить флажки: *Обеспечение целостности данных*, *Каскадное обновление и удаление связанных полей*.
9. Создать однотабличные и двухтабличные формы согласно связям между таблицами. Наполнить таблицы содержательными данными посредством форм. Каждая таблица должна иметь не менее 15 записей – главная таблица, подчиненная таблица – не менее 30 записей.
10. Создать запросы различного типа: на выборку данных с условиями отбора записей, параметрический с двумя параметрами, перекрестный с итоговыми данными, с групповыми операциями (с группировкой данных), с вычислениями.
11. На основе созданных запросов с помощью Мастера создать отчеты, добавить итоги во все отчеты. На основе таблиц создать отчет с групповыми и общими итогами.

### **2.1.2. Требования к содержанию и оформлению отчета**

- Отчет подготовить в MS Word, он должен включать:
- файл с БД;
  - диалоговое окно *Схема данных*;
  - скриншоты созданных объектов;
  - ответы на контрольные вопросы письменно в тетради «вопрос – ответ».

### **2.1.3. Контрольные вопросы**

1. Назовите определение информационной системы.
2. Укажите кратко содержание этапов *логического* и *физического* проектирования базы данных.
4. Дайте понятие *модели* и укажите типы моделей данных.
5. Назовите сущность понятий: *домен*, *кортеж*, *отношение*.
7. Укажите, в чем заключается процедура *нормализации* данных.
8. Дайте определение и укажите смысл понятий «*первичный*» и «*внешний*» ключи базы данных.

10. Перечислите объекты базы данных в MS Access.
11. Назовите способы создания таблиц и их краткое содержание.
12. Назовите способы создания форм.
13. Назовите способы создания запросов.
14. Назовите способы создания отчетов.
15. Назовите типы связей между таблицами.
16. Укажите способы вставки итогов в отчеты.
17. Дайте определение понятиям «главная», «подчиненная» таблица.
18. Раскройте сущность понятия «нормализация» БД.
19. Раскройте сущность понятия «целостность» БД.
20. Укажите последовательность создания отчета с групповыми и общими итогами.

## 2.2. Примеры выполнения лабораторных работ

Ниже приведены примеры созданных объектов: схема данных (физическая модель данных), таблицы, запросы на выборку.

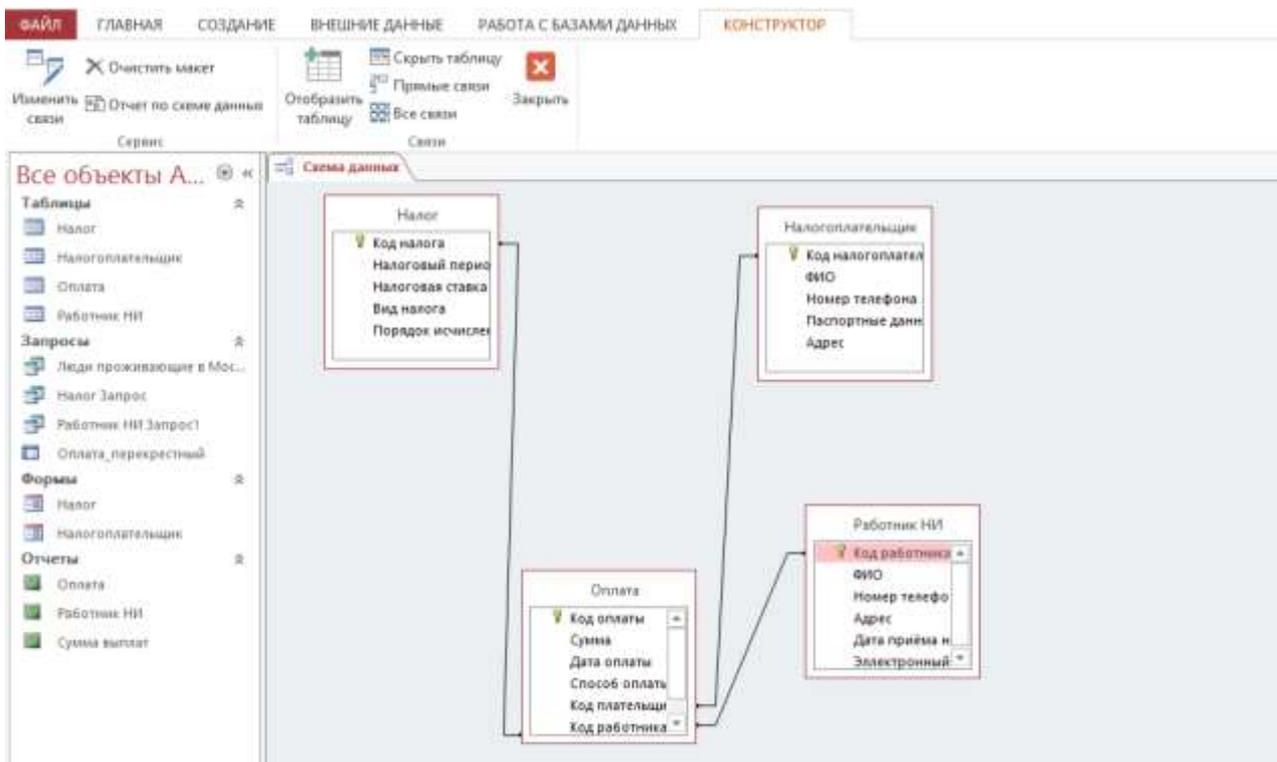


Рисунок 45 – Пример выполнения работы. Структура БД

<span>Схема данных</span> <span>Налог</span> <span>Налогоплательщик</span>					
Код налога	Налоговый период	Налоговая ставка	Вид налога	Порядок исчисления	
1	Месячный (МС)	12%	Прогрессивный	Самим налогоплательщиком	
2	Квартальный (КВ)	16%	Регрессивный	Налоговым агентом	
3	Полугодовой (ПЛ)	10%	Аккордный	Налоговым органом	
4	Годовой (ГД)	12%	Пропорциональный	Налоговым органом	
*	(№)				

Рисунок 46 – Пример выполнения работы. Таблицы. Налог

<span>Схема данных</span> <span>Налог</span> <span>Налогоплательщик</span>					
Код налога	ФИО	Номер телефона	Паспортный	Адрес	
1	Рустамов В. А.	+7-109-789-9879	12 38 653568	Москва	
2	Смирнов Р. Е.	+7-123-456-7890	12 32 468840	Междуреченск	
3	Воробьёв Р. А.	+7-123-456-7890	12 30 969070	Новороссийск	
4	Болотникава А.	+7-123-456-7890	12 34 513673	Катлан	
5	Комова С. А.	+7-123-456-7890	12 34 509631	Москва	

Рисунок 46 – Пример выполнения работы.  
Таблицы. Налогоплательщик

<span>Схема данных</span> <span>Налог</span> <span>Налогоплательщик</span> <span>Оплата</span>						
Код оплаты	Сумма	Дата оплаты	Способ оплаты	Код плател	Код работника	Код налога
1	12 890,00 ₹	12.07.2019	Наличные	2	6	1
2	13 459,00 ₹	19.02.2020	Перевод	3	7	1
3	3 456,00 ₹	25.08.2020	Перевод	4	5	3
4	6 789,00 ₹	24.01.2020	Наличные	1	4	2

Рисунок 47 – Пример выполнения работы. Таблицы. Оплата

<span>Схема данных</span> <span>Налог</span> <span>Налогоплательщик</span> <span>Оплата</span> <span>Работник НИ</span>						
Код работника	ФИО	Номер телефона	Адрес	Дата приёма на	Электронный адрес	
4	Мещеряков П. Р.	+7-123-456-7890	г. Орлов	19.03.2020	masnish@mail.ru	
5	Асланян А. Э.	+7-123-456-7890	г.Калининград	30.12.1999	sunlite@gmail.com	
6	Камкин Д. С.	+7-123-456-7890	г.Москва	26.06.2016	litor@mail.ru	
7	Миноров Р. А.	+7-123-456-7890	г.Краснодар	19.03.2020	hamster@yandex.com	

Рисунок 48 – Пример выполнения работы. Таблицы. Работник НИ

Код налога	ФИО	Номер телефона	Паспортный	Адрес
	Рустамов В. А.	+7-109-789-9879	12 38 653568	Москва
5	Комова С. А.	+7-123-456-7890	12 34 509631	Москва

Рисунок 49 – Пример выполнения работы. Запросы

ФИО	Номер телефона
Мещеряков П. Р.	+7-123-456-7890
Асланян А. Э.	+7-123-456-7890
Камкин Д. С.	+7-123-456-7890
Миноров Р. А.	+7-123-456-7890
*	

Рисунок 50 – Пример выполнения работы. Запросы

Дата оплаты	Код работника	Код плател	Итоговое зп	1	2	3	4
12.07.2019	6	2	12 890,00	12 890,00			
24.01.2020	4	1	6 789,00				6 789,00
19.02.2020	7	3	13 459,00		13 459,00		
25.08.2020	5	4	3 456,00			3 456,00	

Рисунок 51 – Пример выполнения работы. Запросы

### 2.3. Создание основных объектов базы данных

Рассмотрим создание основных объектов базы данных. Таблицы, формы, запросы и отчеты создаются в различных режимах, основные из которых – с помощью мастера или конструктора. Доступ к инструментам осуществляется на вкладке **Создание** (рис. 32). На вкладке присутствует соответствующая группа команд: *Таблицы, Формы, Запросы, Отчеты* (и др.) и способы (режимы) создания.

Запросы – наиболее разнообразный объект БД в смысле различия типов. Поэтому рассмотрим создание запросов различных типов.

## Создание запросов

В окне приложения на вкладке **Создание** (рис. 32) указаны основные способы создания запросов: *Мастер запросов*, *Конструктор запросов* (рис. 52).

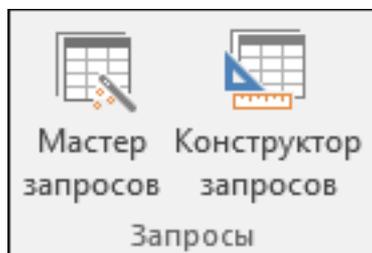


Рисунок 52 – Мастер запросов

Слово запрос (*query*) с латинского языка означает спрашивать, узнавать. Поэтому запрос можно представить как вопрос, имеющий отношение к информации, которая содержится в таблицах БД. Запрос строится на основе одной или нескольких таблиц. При этом могут использоваться как таблицы БД, так и другие запросы.

Запрос является одним из основных инструментов обработки данных в любой СУБД. Он применяется для просмотра, анализа или изменения данных в одной или нескольких таблицах БД. Запросы используются и в качестве источника данных для форм, отчетов, страниц доступа и других запросов.

### Основные типы запросов

Существуют два основных типа запросов:

- *запросы на выборку* выполняют извлечение данных, предназначенных для отображения в виде результирующей таблицы;
- *запросы на выполнение действий* осуществляют вставку, изменение или удаление данных в одной или нескольких таблицах БД.

При выполнении запроса MS Access считывает данные из используемых таблиц, затем отбирает записи, удовлетворяющие заданным условиям отбора, и результат отображает в виде временной таблицы. MS Access не сохраняет выборку, а хранит только инструкцию о том, как должны быть организованы данные в результате выполнения запроса. Поэтому выборка формируется всякий раз заново на основе текущих данных исходных таблиц MS Access.

Через запросы можно выполнить следующие виды обработки данных:

- выбрать записи, удовлетворяющие условиям отбора и отредактировать их;
- получить таблицу с полями, структура которой отлична от исходных таблиц базы данных;
- произвести вычисления в каждой из полученных записей выборки;
- сгруппировать записи с одинаковыми значениями в одном или нескольких полях и выполнить над ними групповые операции;
- создать новую таблицу базы данных, используя данные из существующих таблиц;
- обновить или удалить некоторое подмножество записей из таблицы базы данных;
- добавить подмножество записей в другую таблицу и др.

### Способы создания запросов

В общем случае существуют два способа формирования запросов.

1. *Запросы по образцу* (или QBE-запросы, Query by example – запросы по образцу) или *запросы на выборку данных*, для определения которых пользователь должен указать параметры запроса в окне конструирования, задавая образцы для поиска информации.

2. *Структурированные* или SQL-запросы (Structured Query Language – язык структурированных запросов), для определения которых пользователь должен описать запрос с помощью структурированного языка запросов, используя специальные команды и функции (SELECT, INSERT, UPDATE и DELETE и др.).

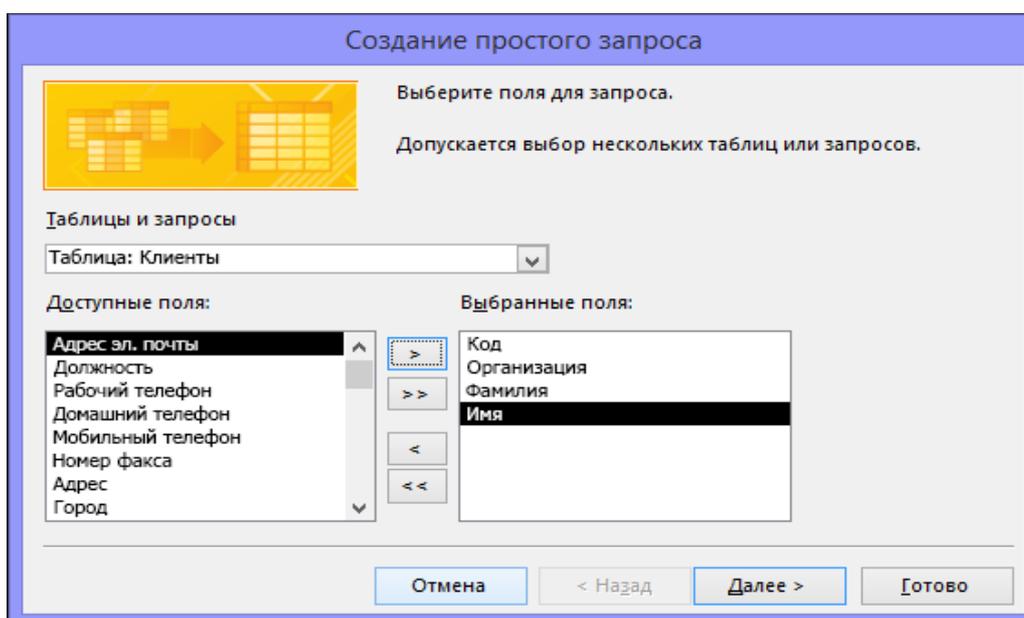


Рисунок 53 – Мастер запросов. Создание простого запроса

Практически любой запрос может быть сформирован как запрос по образцу, при этом MS Access автоматически формирует соответствующий SQL-запрос. Чтобы просмотреть инструкции SQL-запроса, необходимо в списке панели **Конструктора** выбрать пункт *Режим SQL*.

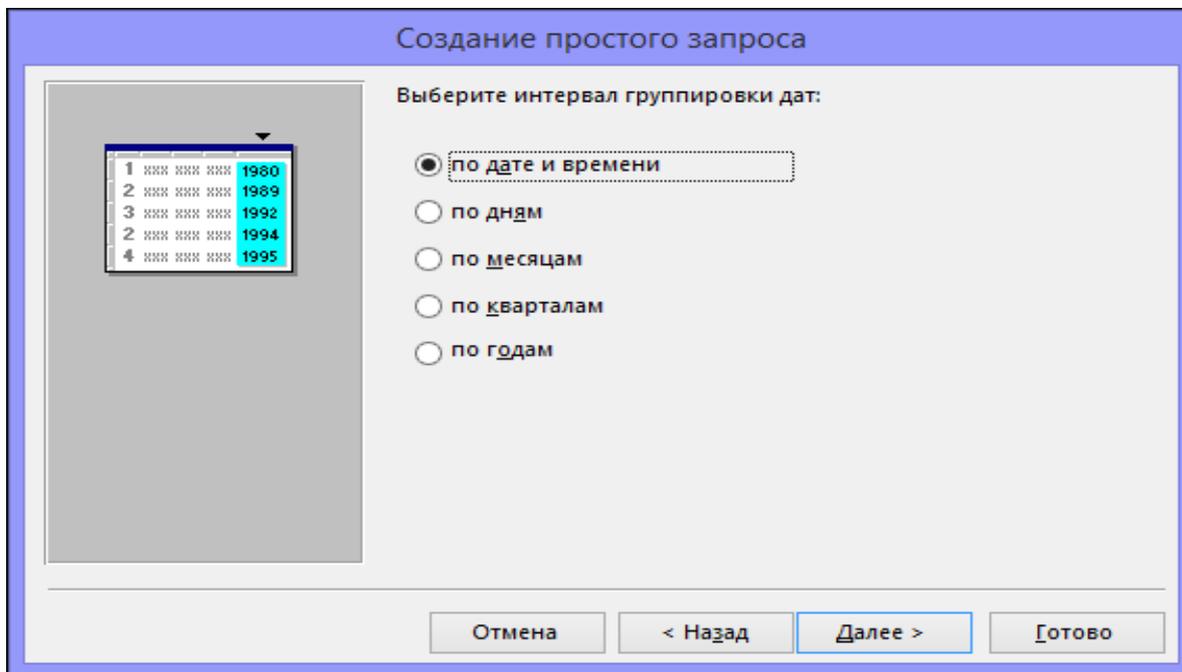


Рисунок 54 – Мастер запросов. Группировка дат

Мастер позволяет создать запрос за конечное число шагов в режиме *диалога с пользователем* (рис. 53, 54).

Режим *конструктора* используется для создания запроса ручным способом или внесения модификаций в ранее созданный запрос.

В режиме *конструктора* для группировки значений по периодам можно использовать выражения, однако в мастере доступны только указанные здесь варианты.

### **Создание запроса на выборку с помощью мастера**

Выполним команду *Создание запроса с помощью мастера* в диалоговом окне (ДО) БД. На первом шаге работы мастера необходимо выбрать источник данных. Запрос может строиться на основе таблицы или другого запроса. В выпадающем списке *Таблицы и запросы* необходимо выбрать требуемую таблицу или запрос. В списке *Доступные поля* отобразятся поля выбранного объекта. Для добавления поля в создаваемый запрос необходимо выделить его и нажать кнопку , для добавления всех полей используется кнопка . Отобранные поля отобразятся в списке *Выбранные поля*. Для

исключения полей из создаваемого запроса используются кнопки  и .

Для добавления полей из нескольких таблиц (запросов) необходимо поочередно добавить поля из каждой таблицы (запроса) описанным выше способом.

В зависимости от выбранных полей количество шагов различно. Если в запросе присутствуют числовые поля, можно выбрать вид представления данных: подробный или итоговый (рис. 55).

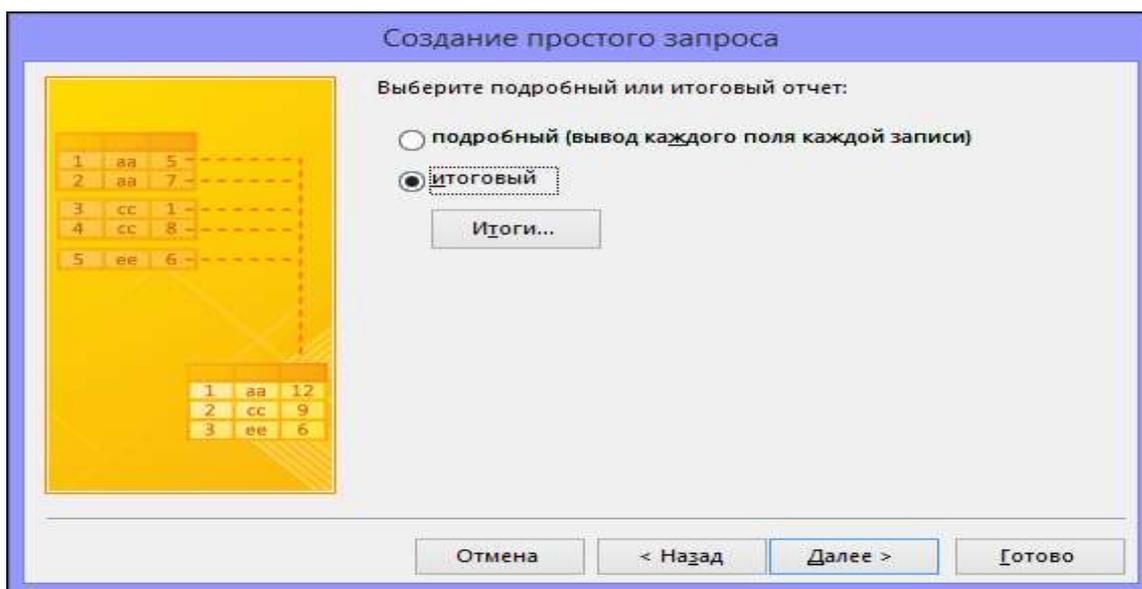


Рисунок 55 – Мастер запросов. Создание простого запроса

При выборе подробного запроса выводятся все записи указанных полей.

При выборе итогового запроса активизируется кнопка **Итоги...**, при нажатии которой открывается ДО для выбора итоговой функции (Sum – сумма, Avg – среднее значение, Min – минимальное значение, Max – максимальное значение). На последнем шаге работы мастера (следует указать имя запроса и дальнейшее действие: открыть запрос в режиме *выполнения* (переключатель  *Открыть запрос для просмотра данных*) или перейти в режим *конструктора* для внесения изменений в макет запроса (переключатель  *Изменить макет запроса*).

После нажатия кнопки **Готово** на закладке **Запросы** ДО БД появится имя созданного запроса.

Если вы решили открыть запрос, он отобразит выбранные данные в режиме *таблицы*. Если вы решили изменить запрос, он откроется в режиме *конструктора*.

## Создание запроса на выборку в режиме конструктора

Для создания запроса в режиме конструктора необходимо в ДО БД выполнить команду *Создание запроса в режиме конструктора*. В результате откроется окно *Конструктора* – бланк запроса (рис. 56).

Автоматически появляется ДО *Добавление таблицы*, в котором необходимо поочередно выбрать нужные (только те, которые требуются для формирования запроса) таблицы (или ранее созданные запросы), используя кнопку **Добавить**. После добавления всех объектов в подсхему данных нажать кнопку **Заккрыть**. Повторно открыть ДО *Добавление таблицы* можно командой *Отобразить таблицу...* в меню *Запрос*, или *Добавить таблицу* из контекстного меню окна конструктора, или кнопкой  на панели инструментов (ПИ) *Конструктор запросов*.

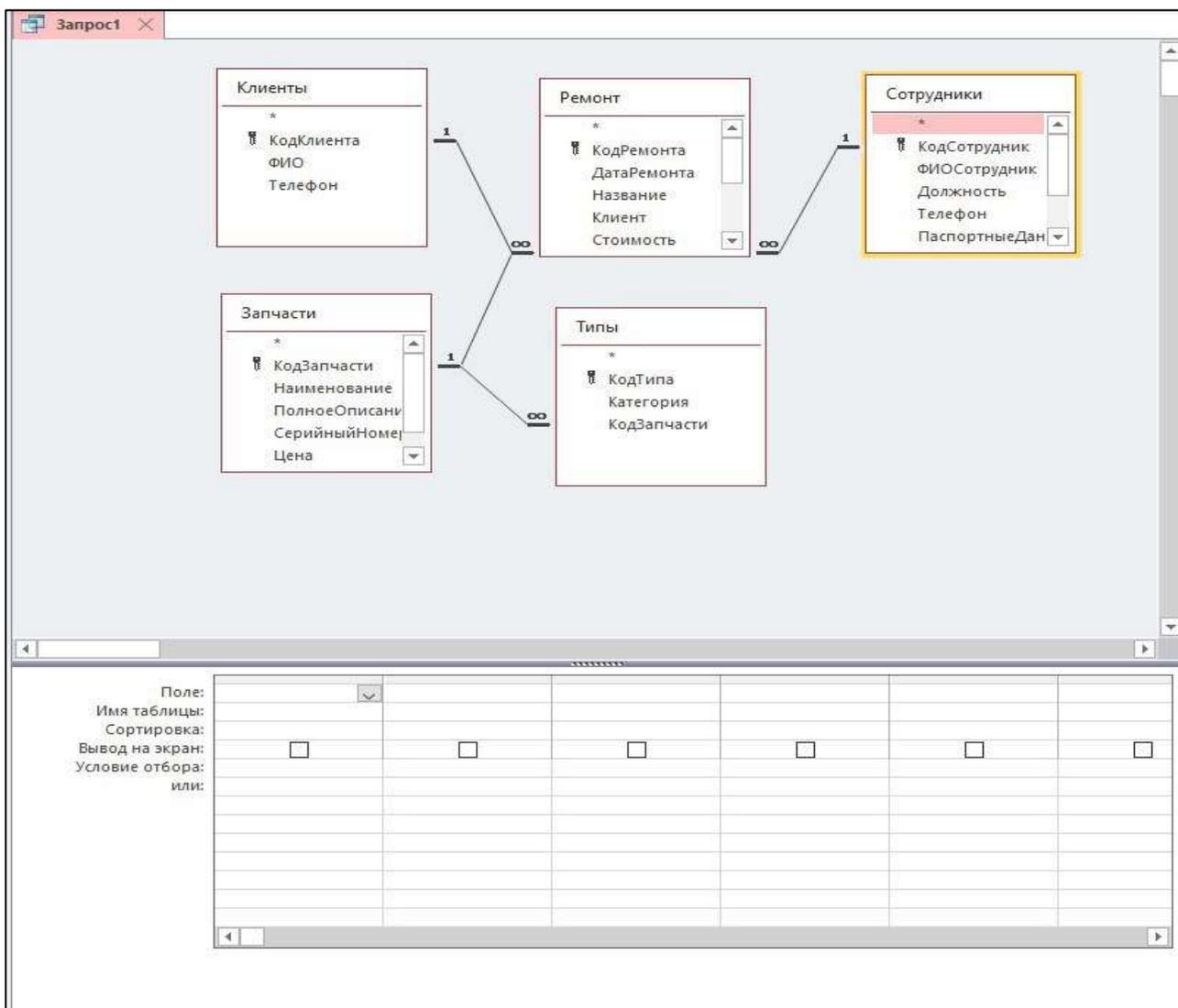


Рисунок 56 – Бланк запросов

Окно *Бланка* запроса разделено на две части: верхняя содержит подсхему данных, нижняя – таблицу бланка.

Связи между объектами схемы данных БД автоматически отображаются в подсхеме данных запроса, недостающие связи следует создать самому.

Для добавления полей в таблицу бланка запроса можно использовать любой из способов:

а) сделать двойной щелчок мышью по имени поля в области подсхемы данных;

б) перетащить имя поля левой кнопкой мыши из подсхемы данных в соответствующий столбец таблицы бланка;

в) выбрать значения из выпадающих списков строк таблицы бланка *Имя таблицы* и *Поле*.

Каждый столбец таблицы бланка запроса содержит параметры настройки одного поля. В таблицу бланка можно добавить до 16 полей и для каждого указать свойства:

– *порядок сортировки* (без сортировки – (отсутствует); по возрастанию; по убыванию);

– *вывод на экран* (если флажок установлен, то поле отображается в результирующей таблице запроса, иначе – не отображается, но участвует в условиях отбора и сортировки);

– *условия отбора* (можно задать несколько критериев отбора записей).

Для удаления поля из таблицы бланка запроса необходимо его выделить и нажать клавишу **Delete**.

Для перемещения поля необходимо выделить столбец и, уцепив мышью за заголовок, перетащить его в нужное место. Все столбцы полей справа от него передвинутся влево.

Для перехода в режим *таблицы* необходимо выполнить команду *Режим таблицы* меню *Вид*, или выполнить команду *Конструктор запросов* контекстного меню окна конструктора запроса, или нажать кнопку  ПИ **Конструктор запросов**.

Для перехода в режим *конструктора* уже открытого запроса необходимо выполнить команду *Конструктор* меню *Вид*, или выполнить эту же команду контекстного меню заголовка окна таблицы запроса, или нажать кнопку  ПИ **Запрос в режиме таблицы**.

## Задание условий отбора записей

Условия для отбора записей можно задать только в режиме конструктора.

Условие отбора – это фильтр, накладываемый на набор записей.

Для задания условий отбора в таблице бланка запроса используются строки (свойства поля) *Условие отбора*, или и все последующие строки для каждого столбца. Условия, расположенные в разных строках для одного или нескольких полей, имеет логическую связку ИЛИ. Для каждого поля можно задать до девяти альтернативных условий. Условия, расположенные для разных полей в одной строке, имеют логическую связку И.

Для задания условия используются выражения – последовательность операндов, соединенных знаками операций. Операндами могут быть константы, параметры, имена функций и др.

В случае использования констант в результирующую таблицу запроса будут отбираться записи, значение полей которых совпадают с указанными значениями констант.

При использовании параметра (параметрический запрос) в результирующую таблицу будут отбираться записи, соответствующие вводимому во время работы с БД пользователем значению параметра (набор отбираемых записей зависит от значения параметра).

Выражения могут содержать литералы, например: «Магазин «Чибис»», #01.01.2009#, 20. При этом текстовые данные заключаются в кавычки, а даты обрамляются символом «#».

Элементы выражения могут быть связаны операторами:

- арифметическими: \*, +, -, /, ^;
- отношения: <, <=, >, >=, <>, =;
- логическими: and, or, not;
- Like (оператор подобия) – используются для сравнения с шаблоном;
- In – для определения входимости элемента в список;
- Between...and... – для выбора значений из интервала.

### Примеры

1) 4 or 5 – выдает все записи, в которых значение соответствующего поля равно 4 или 5.

2) Between 10 and 100 – выдает все записи, в которых значение поля больше или равно 10 и меньше или равно 100.

3) Between #01.01.2009# and #31.03.2009# – выдает все записи, в которых поле содержит дату между 01.01.2009 и 31.03.2009, включая указанные даты.

4) In ("Иванов И.И."; "Петров П.П."; "Сидоров С.С.") – выдает все записи, в которых значение поля равно "Иванов И.И.", или "Петров П.П.", или "Сидоров С.С."

Для задания несовпадающих значений поля используется оператор отрицания Not (логическое НЕ) или знак операции «не равно» (<>). Например, чтобы выбрать записи обо всех контрагентах, кроме «Сбербанк», условие отбора запишется так:

Not «Сбербанк» или <>«Сбербанк».

Иногда поле конкретной записи может не содержать данные, что соответствует значению Null. В MS Access анализ пустых записей полей выполняется с помощью двух операторов: Is Null (пустое значение) и Is Not Null (не пустое значение).

Для поиска записей по шаблону используется оператор Like вида: Like «шаблон». Для конструирования шаблона в качестве символов подстановки используются:

- символ «?» для представления любого одиночного символа из списка (0-9, А-Я, а-я, А-Z, a-z);
- символ «\*» для представления нескольких любых символов;
- символ «#» для представления любой цифры;
- [список] означает, что в данном месте в поле данных может находиться любой из указанных в списке символов;
- [!список] означает, что в указанном месте поля данных может находиться любой символ, который не указан в списке.

Символы подстановки могут повторяться.

### **Примеры**

1) Like «А\*» – выдает все строки, в которых значение поля начинается на букву «А».

2) Like «[!АВВ]\*#» – выдает все строки, в которых значение поля не начинается на буквы «А», «Б» или «В» и заканчивается цифрой.

Для создания сложных условий отбора (с использованием разных полей, арифметических операций, формул, функций и т.д.) лучше использовать специализированный инструмент – построитель выражений (рис. 57).

Построитель выражений доступен для ячеек *Поле* и *Условие отбора*. Он открывается нажатием кнопки  на ПИ **Конструктор запросов** или выбором команды *Построить...* в контекстном меню этих ячеек.

Окно *Построителя выражений* (рис. 57) состоит из трех разделов.

В верхней части окна построителя расположено поле, в котором создается выражение. Допускается непосредственный ввод части выражения в поле выражения или выбор элементов из списков, расположенных ниже.

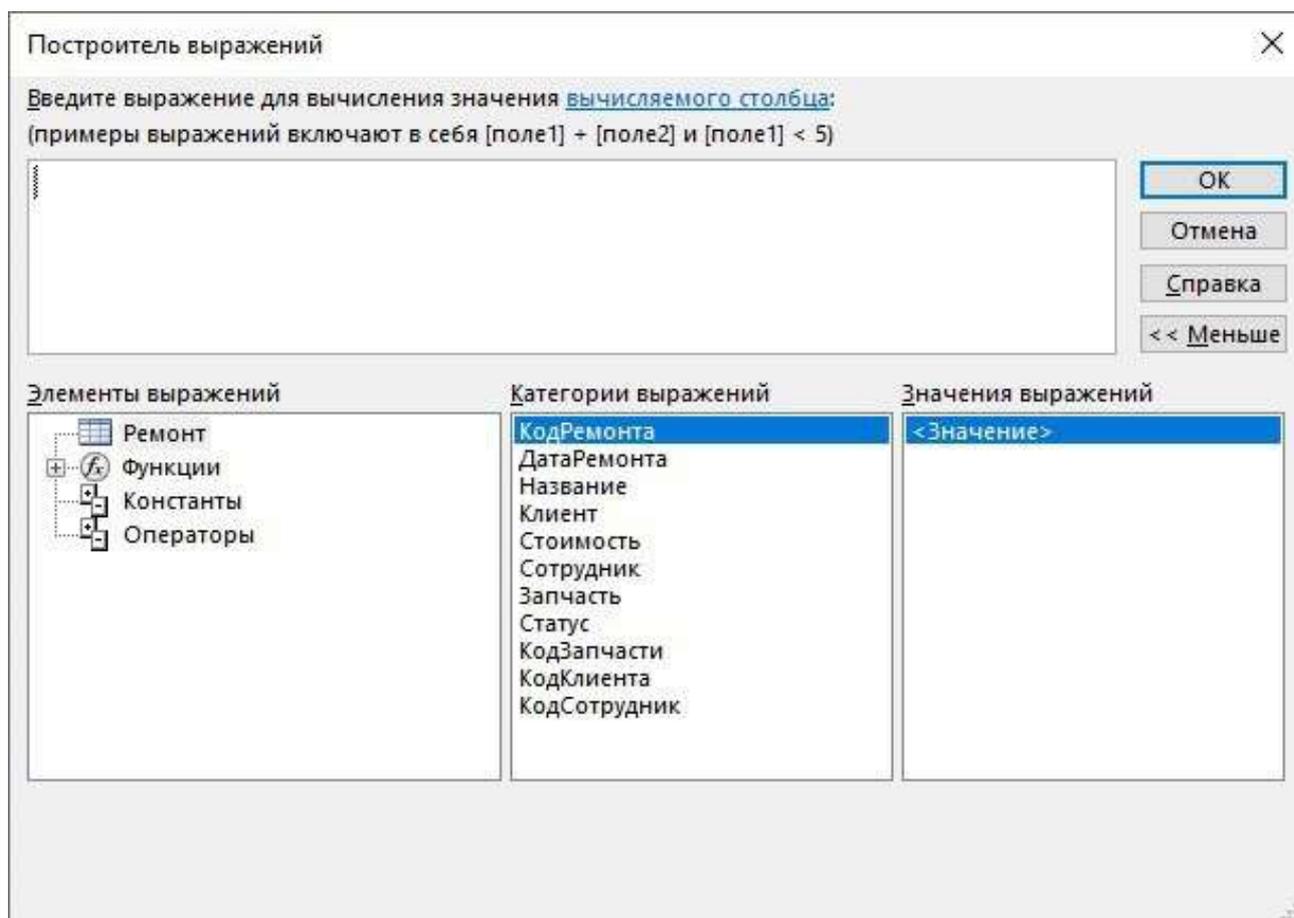


Рисунок 57 – Построитель выражений

В средней части окна построителя находятся кнопки с часто используемыми операторами. При нажатии на одну из этих кнопок построитель вставит соответствующий оператор в текущую позицию поля выражения.

В нижней части окна построителя находятся три поля.

В левом поле выводятся папки, содержащие таблицы, запросы, объекты форм, объекты базы данных, встроенные и определенные пользователем функции, константы, операторы и общие выражения.

В среднем поле задаются определенные элементы или типы элементов для папки, заданной в левом поле. Например, если выбрать в левом поле *Встроенные функции*, то в среднем поле появится список всех типов функций Microsoft Access.

В правом поле выводится список значений (если они существуют) для элементов, заданных в левом и среднем полях. Например, если выбрать в левом поле *Встроенные функции* и указать тип функции в среднем, то в правом поле будет выведен список всех встроенных функций выбранного типа.

Чтобы вывести полный список *Операторов*, необходимо выбрать папку *Операторы* в нижнем левом поле и нужный тип в среднем поле, а в правом поле будут выведены все операторы выбранного типа (рис. 57).

### **Запросы с вычисляемыми полями**

В БД не следует хранить информацию, которую можно вычислить на основе текущих данных. Например, если в таблице хранится информация о количестве и цене проданного товара, то нет необходимости хранить сумму, на которую продан этот товар, т.к. в любой момент ее можно рассчитать. Лишние поля могут привести к неоправданному росту размера БД.

Поля, которые получаются путем вычислений на основе данных других полей, называются *вычисляемыми полями*.

Для создания вычисляемого поля необходимо в конструкторе запроса в строке *Поле:* таблицы бланка запроса в свободном столбце прописать формулу, по которой будут вычисляться значения этого поля. Для указания имени поля следует перед формулой написать заголовок поля и поставить знак двоеточие: *Имя поля: Формула*.

Если имя поля не указано, оно проставляется СУБД автоматически. Для создания вычисляемых полей рекомендуется использовать построитель выражений.

### **Запросы на выполнение действий**

В MS Access существуют следующие типы запросов на выполнение действий:

- запрос на создание таблицы;

- запрос на обновление;
- запрос на удаление;
- запрос на добавление.

Все запросы на выполнение действий создаются в режиме *конструктора*.

Для создания таких запросов вначале формируется обычный запрос на выборку, в который следует поместить все требуемые поля. При этом для запросов на создание таблицы и на добавление можно указать порядок сортировки и условия отбора записей.

Затем выполняется соответствующая команда на изменение типа запроса из меню *Запрос*, или в подменю *Тип запроса* контекстного меню окна конструктора запроса, или в выпадающем списке кнопки **Тип запроса**  (вид кнопки меняется в зависимости от текущего типа запроса: , , , ) на ПИ *Конструктор запроса*.

Для просмотра записей, которые будут обрабатываться запросом на выполнение действия, необходимо перейти в режим *таблицы* (выполнить команду *Режим таблицы* меню *Вид* или другим способом).

Для запуска запроса на выполнение действия в режиме *конструктора* используется команда *Запуск* в меню *Запрос* или кнопка  на ПИ **Конструктор запросов**.

Для запуска запроса на выполнение действия в окне БД достаточно сделать двойной щелчок мышью на его имени на вкладке **Запросы**.

### **Перекрестный запрос**

Перекрестный запрос – это запрос, в котором происходит статистическая обработка данных (подсчитывается сумма, среднее, число значений или выполняются другие расчеты), после чего результаты группируются в виде таблицы по двум наборам данных, один из которых определяет заголовки столбцов, а другой заголовки строк.

Для создания перекрестных запросов необходимо определить три элемента структуры: заглавия строк, заглавия столбцов и область *Значение* (рис. 58).

Для создания перекрестного запроса проще использовать мастер. На вкладке **Запросы** окна БД необходимо нажать кнопку **Создать** и в открывшемся ДО *Новый запрос* выбрать *Перекрестный запрос* (рис. 59).

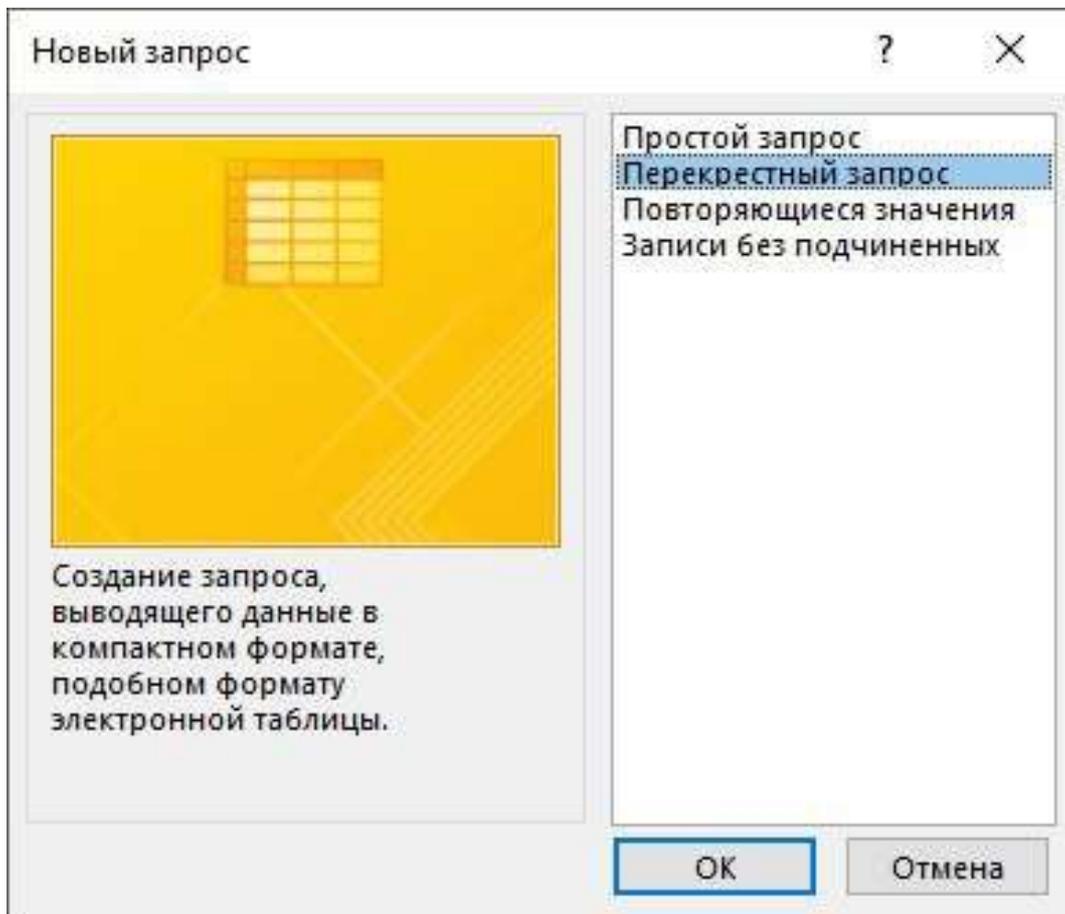


Рисунок 58 – Новый запрос. Перекрестный запрос

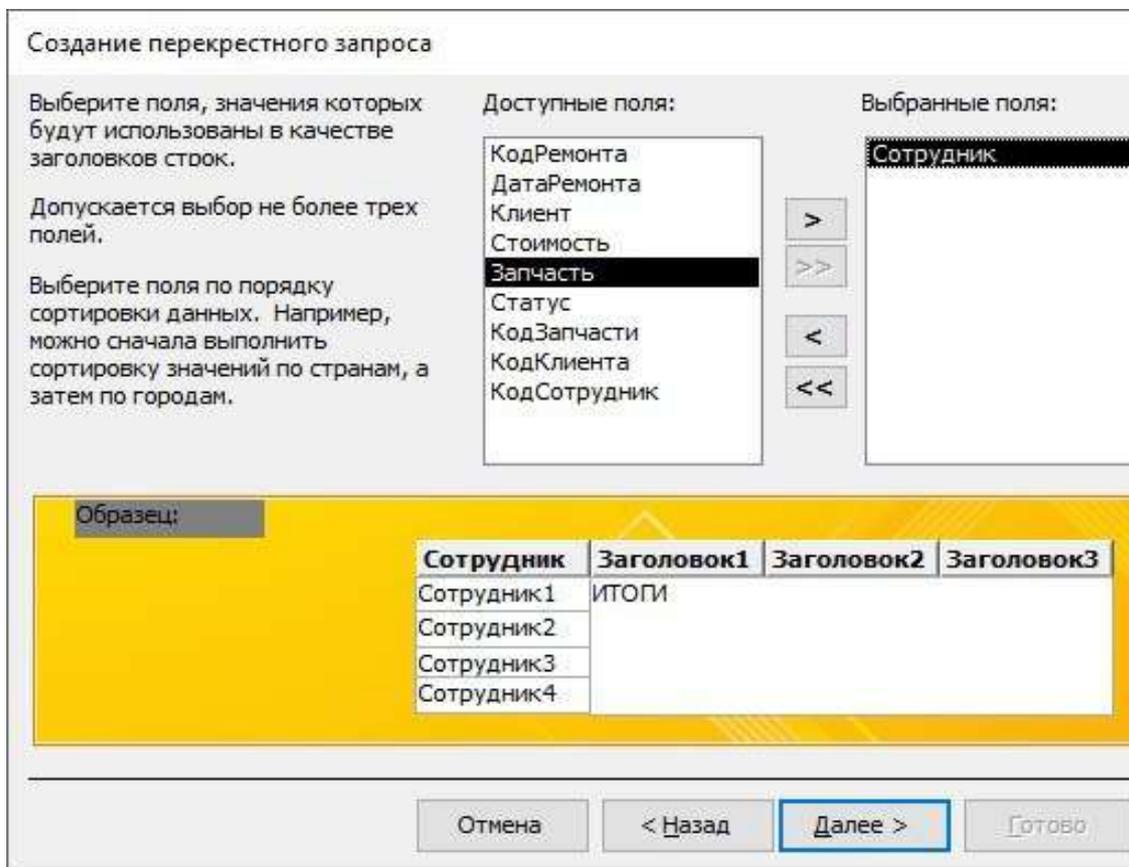


Рисунок 59 – Создание перекрестного запроса

## Список рекомендуемой литературы

1. Таганов, Л. С. Информатика. Решение задач средствами MS Excel : лабораторный практикум для студентов всех специальностей инженерно-экономического и горно-электромеханического факультетов / Л. С. Таганов, А. Г. Пимонов, В. Г. Левин; ГУ КузГТУ. – Кемерово, 2006. – 84 с.
2. Митрофанова, А. Е. Microsoft Excel 2007 : учебное пособие для студентов экономических специальностей / А. Е. Митрофанова, И. А. Соколов; ГУ КузГТУ. – Кемерово, 2010.
3. Бизнес-информатика / О. Е. Иванов, Е. Д. Мещихина, Т. А. Уразаева, А. В. Швецов. – Йошкар-Ола : Поволжский государственный технологический университет, 2018. – 56 с.
4. Асташова, Т. А. Информатика / Т. А. Асташова. – Новосибирск : Изд-во НГТУ, 2017. – 105 с.
5. Мурат, Е. П. Информатика III / Е. П. Мурат. – Ростов-на-Дону; Таганрог : Южный федеральный университет, 2018. – 151 с.
6. Симонович, С. В. Информатика: Базовый курс : учебное пособие для студентов вузов / С. В. Симонович. – Санкт-Петербург : Питер, 2011. – 640 с.
7. Голованов, Н. Н. Геометрическое моделирование : учебник для студентов вузов, обучающихся по направлению «Информатика и вычислительная техника» / Н. Н. Голованов. – Москва : Академия, 2011. – 272 с.
8. Саблина, Г. В. Информатика : учебно-методическое пособие для 1 курса АВТФ направлений 27.03.04 – «Управление в технических системах» и 09.03.01 – «Информатика и ВТ» / Г. В. Саблина. – Новосибирск : Изд-во НГТУ, 2017. – 74 с.
9. Лопатин, В. М. Информатика для инженеров : учебное пособие / В. М. Лопатин. – Санкт-Петербург : Лань, 2019. – 172 с. – ISBN 978-5-8114-3463-3.
10. Гарбер, Г. З. Основы программирования на Visual Basic и VBA в Excel 2007 / Г. З. Гарбер. – Москва : Солон-Пресс, 2008. – 192 с.
11. Гринберг, А. С. Информационные технологии управления : учебное пособие. – Москва: Юнити-Дана, 2015. – 479 с.
12. Матяш, С. А. Информационные технологии управления : курс лекций. – Москва, Берлин : Директ-Медиа, 2014. – 537 с.

13. Информационные технологии : учебник для студентов вузов, обучающихся по специальности 080801 «Прикладная информатика» / О. Л. Голицына [и др.]. – Москва : Форум, 2013. – 608 с.

14. Черников, Б. В. Информационные технологии управления : учебник для студентов вузов, обучающихся по направлениям 080507 «Менеджмент организации» и 080504 «Государственное и муниципальное управление» / Б. В. Черников. – Москва : Форум, 2013. – 368 с.

15. Грошев, А. С. Информационные технологии : лабораторный практикум / А. С. Грошев. – Москва, Берлин : Директ-Медиа, 2015. – 285 с. – Режим доступа:

[http://biblioclub.ru/index.php?page=book\\_red&id=434666](http://biblioclub.ru/index.php?page=book_red&id=434666). – Загл. с экрана. (06.05.2020).

16. Советов, Б. Я. Информационные технологии : учебник для бакалавров: [для студентов вузов, обучающихся по направлениям подготовки диплом. специалистов «Информатика и вычислит. техника» и «Информ. системы»] / Б. Я. Советов, В. В. Цехановский; С.-Петерб. гос. электротехн. ун-т. – Москва : Юрайт, 2012. – 263 с.

17. Информационные технологии в менеджменте (управлении) : учебник и практикум для академического бакалавриата [для студентов вузов, обучающихся по экономическим направлениям и специальностям] / Рос. экон. ун-т им. Г. В. Плеханова; под общ. ред. Ю. Д. Романовой. – Москва : Юрайт, 2015. – 478 с.

18. Советов, Б. Я. Информационные технологии: теоретические основы : учебное пособие для студентов вузов, обучающихся по направлению подготовки бакалавра «Информационные системы и технологии» / Б. Я. Советов, В. В. Цехановский. – Санкт-Петербург : Лань, 2016. – 448 с.

19. Хомоненко А. Д. Базы данных : учебник для высших учебных заведений / А. Д. Хомоненко, В. М. Цыганков, М. Г. Мальцев; под ред. проф. А. Д. Хомоненко. – 5-е изд., доп. – Москва : Бинном-Пресс; Санкт-Петербург : КОРОНА принт, 2006. – 736 с.

20. Робинсон, С. Microsoft Access 2000 : учебный курс / С. Робинсон. – Санкт-Петербург : Питер, 2001. – 512 с.

21. Боровиков, В. В. Microsoft Access 2002. Программирование и разработка баз данных и приложений / В. В. Боровиков. – Москва : СОЛОН-Р, 2002. – 560 с.

22. Каратыгин, С. А. Access 2000. Руководство пользователя с примерами / С. А. Каратыгин. – Москва : Лаборатория базовых знаний, 2000. – 376 с.

23. Федорова, Г. Н. Основы проектирования баз данных : учебное пособие для сред. проф. образования по специальности «Информ. системы (по отраслям)» / Г. Н. Федорова. – Москва : Академия, 2014. – 219 с.: ил.

Глебова Екатерина Андреевна  
Крюкова Валентина Валентиновна

**ИНФОРМАТИКА**  
**ЛАБОРАТОРНЫЙ ПРАКТИКУМ**  
**Учебное пособие**

Печатается в авторской редакции

Подписано в печать 09.07.2020. Формат 60×84/16  
Бумага офсетная. Гарнитура «Times New Roman»  
Уч.-изд. л. 4,7. Тираж 100 экз. Заказ  
Кузбасский государственный технический университет  
имени Т. Ф. Горбачева, 650000, Кемерово, ул. Весенняя, 28  
Издательский центр УИП Кузбасского государственного технического  
университета имени Т. Ф. Горбачева, 650000, Кемерово, ул. Д. Бедного, 4а